Contents I

1 Introduction
   - Problems
   - Types of Communication
   - Notations
   - Basics

2 Broadcast
   - Lower Bound
   - First Results
   - Trees
   - First Results
   - CCC
   - SE
   - BF
   - DB
   - Lower Bounds
   - Disjoint Path Modes

3 Gossip
   - Results
   - Introduction
   - First Results
   - Lines
   - Trees
   - Cycles
   - HQ
   - Hypercube
   - CCC and BF
   - Telephone-Mode
   - Odd Number of Nodes
   - Telegraph-Mode
   - Lower Bound
   - Summary
   - Telegraph-Mode
   - Edge Disjoint Path Mode
Definition of a Broadcasts and Accumulation

**Definition of Broadcast:**
Given are $G = (V, E)$ and $v \in V$.
- $v$ has information $I(v)$
- no node from $V \setminus \{v\}$ knows $I(v)$.
- Each node of $V \setminus \{v\}$ has to receive information $I(v)$.

**Definition of Accumulation:**
Given are $G = (V, E)$ and $v \in V$.
- Each node of $w \in V$ has information $I(w)$
- no node from $V \setminus \{w\}$ knows $I(w)$.
- Node $v$ should receive the information $\cup_{w \in V} I(w)$. 
Definition of a Broadcasts and Accumulation

**Definition of Broadcast:**

Given are $G = (V, E)$ and $v \in V$.

- $v$ has information $I(v)$ and
- no node from $V \setminus \{v\}$ knows $I(v)$.
- Each node of $V \setminus \{v\}$ has to receive information $I(v)$.

**Definition of Accumulation:**

Given are $G = (V, E)$ and $v \in V$.

- Each node of $w \in V$ has information $I(w)$
- no node from $V \setminus \{w\}$ knows $I(w)$.
- Node $v$ should receive the information $\bigcup_{w \in V} I(w)$. 
Definition of a Broadcasts and Accumulation

Definition of Broadcast:
Given are $G = (V, E)$ and $v \in V$.
- $v$ has information $I(v)$ and
- no node from $V \setminus \{v\}$ knows $I(v)$.
- Each node of $V \setminus \{v\}$ has to receive information $I(v)$.

Definition of Accumulation:
Given are $G = (V, E)$ and $v \in V$.
- Each node of $w \in V$ has information $I(w)$
- no node from $V \setminus \{w\}$ knows $I(w)$.
- Node $v$ should receive the information $\bigcup_{w \in V} I(w)$. 
Definition of a Broadcasts and Accumulation

**Definition of Broadcast:**

Given are $G = (V, E)$ and $v \in V$.
- $v$ has information $I(v)$ and
- no node from $V \setminus \{v\}$ knows $I(v)$.
- Each node of $V \setminus \{v\}$ has to receive information $I(v)$.

**Definition of Accumulation:**

Given are $G = (V, E)$ and $v \in V$.
- Each node of $w \in V$ has information $I(w)$
- no node from $V \setminus \{w\}$ knows $I(w)$.
- Node $v$ should receive the information $\bigcup_{w \in V} I(w)$.
Definition of a Broadcasts and Accumulation

**Definition of Broadcast:**

Given are $G = (V, E)$ and $v \in V$.

- $v$ has information $I(v)$ and
- no node from $V \setminus \{v\}$ knows $I(v)$.
- Each node of $V \setminus \{v\}$ has to receive information $I(v)$.

**Definition of Accumulation:**

Given are $G = (V, E)$ and $v \in V$.

- Each node of $w \in V$ has information $I(w)$
- no node from $V \setminus \{w\}$ knows $I(w)$.
- Node $v$ should receive the information $\bigcup_{w \in V} I(w)$. 
Definition of a Broadcasts and Accumulation

Definition of Broadcast:
Given are $G = (V, E)$ and $v \in V$.
- $v$ has information $I(v)$ and
- no node from $V \setminus \{v\}$ knows $I(v)$.
- Each node of $V \setminus \{v\}$ has to receive information $I(v)$.

Definition of Accumulation:
Given are $G = (V, E)$ and $v \in V$.
- Each node of $w \in V$ has information $I(w)$ and
- no node from $V \setminus \{w\}$ knows $I(w)$.
- Node $v$ should receive the information $\bigcup_{w \in V} I(w)$. 
Definition of Broadcast:

Given are $G = (V, E)$ and $v \in V$.

- $v$ has information $I(v)$ and
- no node from $V \setminus \{v\}$ knows $I(v)$.
- Each node of $V \setminus \{v\}$ has to receive information $I(v)$.

Definition of Accumulation:

Given are $G = (V, E)$ and $v \in V$.

- Each node of $w \in V$ has information $I(w)$ and
- no node from $V \setminus \{w\}$ knows $I(w)$.
- Node $v$ should receive the information $\bigcup_{w \in V} I(w)$. 
Definition of a Gossip

Definition of Accumulation:

Given are \( G = (V, E) \) and \( v \in V \).

- Each node of \( w \in V \) has information \( I(w) \).
- No node from \( V \setminus \{w\} \) knows \( I(w) \).
- Node \( v \) should receive the information \( \bigcup_{w \in V} I(w) \).

Definition (Gossip):

Given is \( G = (V, E) \).

- Each node of \( w \in V \) has information \( I(w) \).
- No node from \( V \setminus \{w\} \) knows \( I(w) \).
- Each node of \( v \in V \) should receive the information \( \bigcup_{w \in V} I(w) \).
Definition of a Gossip

**Definition of Accumulation:**

Given are $G = (V, E)$ and $v \in V$.
- Each node of $w \in V$ has information $I(w)$
- no node from $V \setminus \{w\}$ knows $I(w)$.
- Node $v$ should receive the information $\bigcup_{w \in V} I(w)$.

**Definition (Gossip):**

Given is $G = (V, E)$.
- Each node of $w \in V$ has information $I(w)$
- no node from $V \setminus \{w\}$ knows $I(w)$.
- Each node of $v \in V$ should receive the information $\bigcup_{w \in V} I(w)$.
Definition of a Gossip

Definition of Accumulation:

Given are $G = (V, E)$ and $v \in V$.

- Each node of $w \in V$ has information $I(w)$
- no node from $V \setminus \{w\}$ knows $I(w)$.
- Node $v$ should receive the information $\bigcup_{w \in V} I(w)$.

Definition (Gossip):

Given is $G = (V, E)$.

- Each node of $w \in V$ has information $I(w)$
- no node from $V \setminus \{w\}$ knows $I(w)$.
- Each node of $v \in V$ should receive the information $\bigcup_{w \in V} I(w)$.
Definition of a Gossip

Definition of Accumulation:
Given are $G = (V, E)$ and $v \in V$.

- Each node of $w \in V$ has information $I(w)$
- No node from $V \setminus \{w\}$ knows $I(w)$.
- Node $v$ should receive the information $\bigcup_{w \in V} I(w)$.

Definition (Gossip):
Given is $G = (V, E)$.

- Each node of $w \in V$ has information $I(w)$
- No node from $V \setminus \{w\}$ knows $I(w)$.
- Each node of $v \in V$ should receive the information $\bigcup_{w \in V} I(w)$. 
Definition of a Gossip

Definition of Accumulation:

Given are $G = (V, E)$ and $v \in V$.

- Each node of $w \in V$ has information $I(w)$
- no node from $V \setminus \{w\}$ knows $I(w)$.
- Node $v$ should receive the information $\bigcup_{w \in V} I(w)$.

Definition (Gossip):

Given is $G = (V, E)$.

- Each node of $w \in V$ has information $I(w)$
- no node from $V \setminus \{w\}$ knows $I(w)$.
- Each node of $v \in V$ should receive the information $\bigcup_{w \in V} I(w)$. 
Definition of a Gossip

Definition of Accumulation:

Given are $G = (V, E)$ and $v \in V$.

- Each node of $w \in V$ has information $I(w)$ and
- no node from $V \setminus \{w\}$ knows $I(w)$.
- Node $v$ should receive the information $\bigcup_{w \in V} I(w)$.

Definition (Gossip):

Given is $G = (V, E)$.

- Each node of $w \in V$ has information $I(w)$ and
- no node from $V \setminus \{w\}$ knows $I(w)$.
- Each node of $v \in V$ should receive the information $\bigcup_{w \in V} I(w)$.
Definition of a Gossip

**Definition of Accumulation:**
Given are $G = (V, E)$ and $v \in V$.
- Each node of $w \in V$ has information $I(w)$ and
- no node from $V \setminus \{w\}$ knows $I(w)$.
- Node $v$ should receive the information $\cup_{w \in V} I(w)$.

**Definition (Gossip):**
Given is $G = (V, E)$.
- Each node of $w \in V$ has information $I(w)$ and
- no node from $V \setminus \{w\}$ knows $I(w)$.
- Each node of $v \in V$ should receive the information $\cup_{w \in V} I(w)$.
Types of Communication

- **Telegraph-Mode**: Communication is directed.
  - Is also called one-way communication.
- **Telephone-Mode**: Information is exchanged.
  - Is also called two-way communication.
  - Communication only between neighbours.
  - Communication is done in rounds.
  - In each round the active edges are a matching.
  - Each round uses one time-unit.
Types of Communication

- **Telegraph-Mode**: Communication is directed.
  - Is also called one-way communication.
- **Telephone-Mode**: Information is exchanged.
  - Is also called two-way communication.
- Communication only between neighbours.
- Communication is done in rounds.
- In each round the active edges are a matching.
- Each round uses one time-unit.
Types of Communication

- Telegraph-Mode: Communication is directed.
  - Is also called one-way communication.
- Telephone-Mode: Information is exchanged.
  - Is also called two-way communication.
  - Communication only between neighbours.
  - Communication is done in rounds.
  - In each round the active edges are a matching.
  - Each round uses one time-unit.
Types of Communication

- Telegraph-Mode: Communication is directed.
  - Is also called one-way communication.
- Telephone-Mode: Information is exchanged.
  - Is also called two-way communication.
  - Communication only between neighbours.
  - Communication is done in rounds.
  - In each round the active edges are a matching.
  - Each round uses one time-unit.
Types of Communication

- **Telegraph-Mode**: Communication is directed.
  - Is also called one-way communication.
- **Telephone-Mode**: Information is exchanged.
  - Is also called two-way communication.
- **Communication only between neighbours.**
  - Communication is done in rounds.
  - In each round the active edges are a matching.
  - Each round uses one time-unit.
Types of Communication

- **Telegraph-Mode**: Communication is directed.
  - Is also called one-way communication.
- **Telephone-Mode**: Information is exchanged.
  - Is also called two-way communication.
- Communication only between neighbours.
- **Communication is done in rounds**.
  - In each round the active edges are a matching.
  - Each round uses one time-unit.
Types of Communication

- **Telegraph-Mode**: Communication is directed.
  - Is also called one-way communication.
- **Telephone-Mode**: Information is exchanged.
  - Is also called two-way communication.
- Communication only between neighbours.
- Communication is done in rounds.
- **In each round the active edges are a matching.**
- Each round uses one time-unit.
Types of Communication

- **Telegraph-Mode**: Communication is directed.
  - Is also called one-way communication.
- **Telephone-Mode**: Information is exchanged.
  - Is also called two-way communication.
- Communication only between neighbours.
- Communication is done in rounds.
- In each round the active edges are a matching.
- Each round uses one time-unit.
**Types of Communication**

- In the broadcast-problem the information of one node is transferred to all others.
- The accumulation-problem is a “inverse” broadcast.
- A gossip distributes the sum of all informations to all nodes.
- In each round the communication is done by a matching.
- The communication on an edge may be one-way or two-way, depending on the mode.
- The size of send date is ignored.
Types of Communication

- In the broadcast-problem the information of one node is transferred to all others.

- The accumulation-problem is a “inverse” broadcast.

- A gossip distributes the sum of all informations to all nodes.

- In each round the communication is done by a matching.

- The communication on an edge may be one-way or two-way, depending on the mode.

- The size of send date is ignored.
Types of Communication

- In the broadcast-problem the information of one node is transferred to all others.
- The accumulation-problem is a “inverse” broadcast.
- A gossip distributes the sum of all informations to all nodes.
- In each round the communication is done by a matching.
- The communication on an edge may be one-way or two-way, depending on the mode.
- The size of send date is ignored.
Types of Communication

- In the broadcast-problem the information of one node is transfered to all others.
- The accumulation-problem is a “inverse” broadcast.
- A gossip distributes the sum of all informations to all nodes.
- In each round the communication is done by a matching.
- The communication on an edge may be one-way or two-way, depending on the mode.
- The size of send date is ignored.
Types of Communication

- In the broadcast-problem the information of one node is transferred to all others.
- The accumulation-problem is a “inverse” broadcast.
- A gossip distributes the sum of all informations to all nodes.
- In each round the communication is done by a matching.
- The communication on an edge may be one-way or two-way, depending on the mode.
- The size of send date is ignored.
Types of Communication

- In the broadcast-problem the information of one node is transferred to all others.
- The accumulation-problem is a “inverse” broadcast.
- A gossip distributes the sum of all informations to all nodes.
- In each round the communication is done by a matching.
- The communication on an edge may be one-way or two-way, depending on the mode.
- The size of send date is ignored.
Definition

- By $comm(A)$ we denote the complexity (number of rounds) of a communication-algorithm.

- $r(G) = \min\{comm(A) \mid A \text{ is a one-way algorithm for the gossip-problem on } G\}$

- $r_2(G) = \min\{comm(A) \mid A \text{ is a two-way algorithm for the gossip-problem on } G\}$

- $b(v, G) = \min\{comm(A) \mid A \text{ is a one-way algorithm for the broadcast-problem on } G \text{ and } v\}$

- $a(v, G) = \min\{comm(A) \mid A \text{ is a one-way algorithm for the accumulations-problem on } G \text{ and } v\}$
Definition

- By $\text{comm}(A)$ we denote the complexity (number of rounds) of a communication-algorithm.

- $r(G) = \min \{ \text{comm}(A) \mid A \text{ is a one-way algorithm for the gossip-problem on } G \}$

- $r_2(G) = \min \{ \text{comm}(A) \mid A \text{ is a two-way algorithm for the gossip-problem on } G \}$

- $b(v, G) = \min \{ \text{comm}(A) \mid A \text{ is a one-way algorithm for the broadcast-problem on } G \text{ and } v \}$

- $a(v, G) = \min \{ \text{comm}(A) \mid A \text{ is a one-way algorithm for the accumulations-problem on } G \text{ and } v \}$
Definition

- By \( \text{comm}(A) \) we denote the complexity (number of rounds) of a communication-algorithm.

- \( r(G) = \min \{ \text{comm}(A) \mid A \text{ is a one-way algorithm for the gossip-problem on } G \} \)

- \( r_2(G) = \min \{ \text{comm}(A) \mid A \text{ is a two-way algorithm for the gossip-problem on } G \} \)

- \( b(v, G) = \min \{ \text{comm}(A) \mid A \text{ is a one-way algorithm for the broadcast-problem on } G \text{ and } v \} \)

- \( a(v, G) = \min \{ \text{comm}(A) \mid A \text{ is a one-way algorithm for the accumulations-problem on } G \text{ and } v \} \)
Definition

- By $\text{comm}(A)$ we denote the complexity (number of rounds) of a communication-algorithm.

- $r(G) = \min\{\text{comm}(A) \mid A \text{ is a one-way algorithm for the gossip-problem on } G\}$

- $r_2(G) = \min\{\text{comm}(A) \mid A \text{ is a two-way algorithm for the gossip-problem on } G\}$

- $b(v, G) = \min\{\text{comm}(A) \mid A \text{ is a one-way algorithm for the broadcast-problem on } G \text{ and } v\}$

- $a(v, G) = \min\{\text{comm}(A) \mid A \text{ is a one-way algorithm for the accumulations-problem on } G \text{ and } v\}$
Definition

- By \( \text{comm}(A) \) we denote the complexity (number of rounds) of a communication-algorithm.

- \( r(G) = \min \{ \text{comm}(A) \mid A \text{ is a one-way algorithm for the gossip-problem on } G \} \)

- \( r_2(G) = \min \{ \text{comm}(A) \mid A \text{ is a two-way algorithm for the gossip-problem on } G \} \)

- \( b(v, G) = \min \{ \text{comm}(A) \mid A \text{ is a one-way algorithm for the broadcast-problem on } G \text{ and } v \} \)

- \( a(v, G) = \min \{ \text{comm}(A) \mid A \text{ is a one-way algorithm for the accumulations-problem on } G \text{ and } v \} \)
First Results

- For each graph $G$ and $v \in V$ we have:
  - $a(v, G) = b(v, G)$
  - $a(G) = b(G)$
  - $\text{mina}(G) = \text{minb}(G)$

- Note: reverse broadcast is accumulation.
- There exists a graph $G$ with: $r(G) = 2 \cdot r_2(G)$.
  - Note: 2-clique or cycle of length four.
- The following holds: $\text{minb}(G) \leq b(G) \leq r_2(G) \leq r(G) \leq 2 \cdot r_2(G)$.
- The inequalities result from the definitions.
- $\text{minb}(L(n)) = \lceil n/2 \rceil$
- Optimal broadcast on a line start in the center of the line.
- $b(L(n)) = n - 1$
- A message from the left has to traverse all edges.
First Results

- For each graph $G$ and $v \in V$ we have:
  - $a(v, G) = b(v, G)$
  - $a(G) = b(G)$
  - $\text{mina}(G) = \text{minb}(G)$

  Note: reverse broadcast is accumulation.

- There exists a graph $G$ with: $r(G) = 2 \cdot r_2(G)$.

  Note: 2-clique or cycle of length four.

- The following holds: $\text{minb}(G) \leq b(G) \leq r_2(G) \leq r(G) \leq 2 \cdot r_2(G)$.

  The inequalities result from the definitions.

- $\text{minb}(L(n)) = \lceil n/2 \rceil$

- Optimal broadcast on a line start in the center of the line.

- $b(L(n)) = n - 1$

  A message from the left has to traverse all edges.
First Results

- For each graph $G$ and $v \in V$ we have:
  - $a(v, G) = b(v, G)$
  - $a(G) = b(G)$
  - $\text{mina}(G) = \text{minb}(G)$

- Note: reverse broadcast is accumulation.
- There exists a graph $G$ with: $r(G) = 2 \cdot r_2(G)$.
- Note: 2-clique or cycle of length four.
- The following holds: $\text{minb}(G) \leq b(G) \leq r_2(G) \leq r(G) \leq 2 \cdot r_2(G)$.
- The inequalities result from the definitions.
- $\text{minb}(L(n)) = \lceil n/2 \rceil$
- Optimal broadcast on a line start in the center of the line.
- $b(L(n)) = n - 1$
- A message from the left has to traverse all edges.
First Results

- For each graph $G$ and $v \in V$ we have:
  - $a(v, G) = b(v, G)$
  - $a(G) = b(G)$
  - $\text{mina}(G) = \text{minb}(G)$

- Note: reverse broadcast is accumulation.
- There exists a graph $G$ with: $r(G) = 2 \cdot r_2(G)$.
- Note: 2-clique or cycle of length four.
- The following holds: $\text{minb}(G) \leq b(G) \leq r_2(G) \leq r(G) \leq 2 \cdot r_2(G)$.
- The inequalities result from the definitions.
- $\text{minb}(L(n)) = \lceil n/2 \rceil$
- Optimal broadcast on a line start in the center of the line.
- $b(L(n)) = n - 1$
- A message from the left has to traverse all edges.
First Results

- For each graph $G$ and $v \in V$ we have:
  - $a(v, G) = b(v, G)$
  - $a(G) = b(G)$
  - $\text{mina}(G) = \text{minb}(G)$

- Note: reverse broadcast is accumulation.

- There exists a graph $G$ with: $r(G) = 2 \cdot r_2(G)$.
- Note: 2-clique or cycle of length four.

- The following holds: $\text{minb}(G) \leq b(G) \leq r_2(G) \leq r(G) \leq 2 \cdot r_2(G)$.
- The inequalities result from the definitions.

- $\text{minb}(L(n)) = \lceil n/2 \rceil$

- Optimal broadcast on a line start in the center of the line.

- $b(L(n)) = n - 1$

- A message from the left has to traverse all edges.
First Results

- For each graph $G$ and $v \in V$ we have:
  - $a(v, G) = b(v, G)$
  - $a(G) = b(G)$
  - $\text{mina}(G) = \text{minb}(G)$

  - Note: reverse broadcast is accumulation.

- There exists a graph $G$ with: $r(G) = 2 \cdot r_2(G)$.

  - Note: 2-clique or cycle of length four.

- The following holds: $\text{minb}(G) \leq b(G) \leq r_2(G) \leq r(G) \leq 2 \cdot r_2(G)$.

- The inequalities result from the definitions.

- $\text{minb}(L(n)) = \lceil n/2 \rceil$

- Optimal broadcast on a line start in the center of the line.

- $b(L(n)) = n - 1$

- A message from the left has to traverse all edges.
First Results

- For each graph $G$ and $v \in V$ we have:
  - $a(v, G) = b(v, G)$
  - $a(G) = b(G)$
  - $\text{mina}(G) = \text{minb}(G)$

- Note: reverse broadcast is accumulation.
- There exists a graph $G$ with: $r(G) = 2 \cdot r_2(G)$.
- Note: 2-clique or cycle of length four.
- The following holds: $\text{minb}(G) \leq b(G) \leq r_2(G) \leq r(G) \leq 2 \cdot r_2(G)$.
- The inequalities result from the definitions.
- $\text{minb}(L(n)) = \lceil n/2 \rceil$
- Optimal broadcast on a line start in the center of the line.
- $b(L(n)) = n - 1$
- A message from the left has to traverse all edges.
First Results

- For each graph $G$ and $v \in V$ we have:
  - $a(v, G) = b(v, G)$
  - $a(G) = b(G)$
  - $\text{mina}(G) = \text{minb}(G)$

- Note: reverse broadcast is accumulation.

- There exists a graph $G$ with: $r(G) = 2 \cdot r_2(G)$.

- Note: 2-clique or cycle of length four.

- The following holds: $\text{minb}(G) \leq b(G) \leq r_2(G) \leq r(G) \leq 2 \cdot r_2(G)$.

- The inequalities result from the definitions.

- $\text{minb}(L(n)) = \lceil n/2 \rceil$

- Optimal broadcast on a line start in the center of the line.

- $b(L(n)) = n - 1$

- A message from the left has to traverse all edges.
First Results

- For each graph $G$ and $v \in V$ we have:
  - $a(v, G) = b(v, G)$
  - $a(G) = b(G)$
  - $\text{mina}(G) = \text{minb}(G)$

- Note: reverse broadcast is accumulation.

- There exists a graph $G$ with: $r(G) = 2 \cdot r_2(G)$.
  - Note: 2-clique or cycle of length four.

- The following holds: $\text{minb}(G) \leq b(G) \leq r_2(G) \leq r(G) \leq 2 \cdot r_2(G)$.

- The inequalities result from the definitions.

- $\text{minb}(L(n)) = \lceil n/2 \rceil$

- Optimal broadcast on a line start in the center of the line.

- $b(L(n)) = n - 1$

- A message from the left has to traverse all edges.
First Results

- For each graph $G$ and $v \in V$ we have:
  - $a(v, G) = b(v, G)$
  - $a(G) = b(G)$
  - $\text{mina}(G) = \text{minb}(G)$

- Note: reverse broadcast is accumulation.

- There exists a graph $G$ with: $r(G) = 2 \cdot r_2(G)$.
- Note: 2-clique or cycle of length four.

- The following holds: $\text{minb}(G) \leq b(G) \leq r_2(G) \leq r(G) \leq 2 \cdot r_2(G)$.

- The inequalities result from the definitions.

- $\text{minb}(L(n)) = \lceil n/2 \rceil$

- Optimal broadcast on a line start in the center of the line.

- $b(L(n)) = n - 1$

- A message from the left has to traverse all edges.
First Results

For each graph $G$ and $v \in V$ we have:

- $a(v, G) = b(v, G)$
- $a(G) = b(G)$
- $\text{mina}(G) = \text{minb}(G)$

Note: reverse broadcast is accumulation.

There exists a graph $G$ with: $r(G) = 2 \cdot r_2(G)$.

Note: 2-clique or cycle of length four.

The following holds: $\text{minb}(G) \leq b(G) \leq r_2(G) \leq r(G) \leq 2 \cdot r_2(G)$.

The inequalities result from the definitions.

$\text{minb}(L(n)) = \lceil n/2 \rceil$

Optimal broadcast on a line start in the center of the line.

$b(L(n)) = n - 1$

A message from the left has to traverse all edges.
First Results

- For each graph $G$ and $v \in V$ we have:
  - $a(v, G) = b(v, G)$
  - $a(G) = b(G)$
  - $\text{mina}(G) = \text{minb}(G)$

- Note: reverse broadcast is accumulation.
- There exists a graph $G$ with: $r(G) = 2 \cdot r_2(G)$.
- Note: 2-clique or cycle of length four.
- The following holds: $\text{minb}(G) \leq b(G) \leq r_2(G) \leq r(G) \leq 2 \cdot r_2(G)$.
- The inequalities result from the definitions.
- $\text{minb}(L(n)) = \lceil n/2 \rceil$
- Optimal broadcast on a line start in the center of the line.
- $b(L(n)) = n - 1$
- A message from the left has to traverse all edges.
First Results

- For each graph $G$ and $v \in V$ we have:
  - $a(v, G) = b(v, G)$
  - $a(G) = b(G)$
  - $\text{mina}(G) = \text{minb}(G)$

- Note: reverse broadcast is accumulation.

- There exists a graph $G$ with: $r(G) = 2 \cdot r_2(G)$.
- Note: 2-clique or cycle of length four.

- The following holds: $\text{minb}(G) \leq b(G) \leq r_2(G) \leq r(G) \leq 2 \cdot r_2(G)$.

- The inequalities result from the definitions.

- $\text{minb}(L(n)) = \lceil n/2 \rceil$

- Optimal broadcast on a line start in the center of the line.

- $b(L(n)) = n - 1$

- A message from the left has to traverse all edges.
First Results II

Lemma:
For each graph $G$ with $|V| \geq 2$ we have:

- $b(G) \leq r(G) \leq 2 \cdot \min b(G)$
- $b(G) \leq r_2(G) \leq 2 \cdot \min b(G) - 1$

Proof: Consider the following steps.

- Let $v \in V$ with $b(v, G) = \min b(G) = \min a(G) = z$.
- Let $A = E_1, E_2, \cdots E_z$ be the corresponding one-way broadcast-algorithm.
- Let $B = F_1, F_2, \cdots F_z$ be the corresponding one-way accumulation-algorithm.
- Then is $F_1, F_2, \cdots F_z, E_1, E_2, \cdots E_z$ one-way gossip-algorithm.
- Note: in the two-way case holds: $F_z = E_1$.
- Note: For $L(2 \cdot n)$ we have equality.
Lemma:
For each graph $G$ with $|V| \geq 2$ we have:

- $b(G) \leq r(G) \leq 2 \cdot \min b(G)$
- $b(G) \leq r_2(G) \leq 2 \cdot \min b(G) - 1$

Proof: Consider the following steps.

- Let $v \in V$ with $b(v, G) = \min b(G) = \mina(G) = z$.
- Let $A = E_1, E_2, \cdots E_z$ be the corresponding one-way broadcast-algorithm.
- Let $B = F_1, F_2, \cdots F_z$ be the corresponding one-way accumulation-algorithm.
- Then is $F_1, F_2, \cdots F_z, E_1, E_2, \cdots E_z$ one-way gossip-algorithm.
- Note: in the two-way case holds: $F_z = E_1$.
- Note: For $L(2 \cdot n)$ we have equality.
First Results II

Lemma:

For each graph $G$ with $|V| \geq 2$ we have:

- $b(G) \leq r(G) \leq 2 \cdot \min b(G)$
- $b(G) \leq r_2(G) \leq 2 \cdot \min b(G) - 1$

Proof: Consider the following steps.

- Let $v \in V$ with $b(v, G) = \min b(G) = \min a(G) = z$.
- Let $A = E_1, E_2, \cdots E_z$ be the corresponding one-way broadcast-algorithm.
- Let $B = F_1, F_2, \cdots F_z$ be the corresponding one-way accumulation-algorithm.
- Then is $F_1, F_2, \cdots F_z, E_1, E_2, \cdots E_z$ one-way gossip-algorithm.
- Note: in the two-way case holds: $F_z = E_1$.
- Note: For $L(2 \cdot n)$ we have equality.
First Results II

Lemma:

For each graph $G$ with $|V| \geq 2$ we have:

- $b(G) \leq r(G) \leq 2 \cdot \min b(G)$
- $b(G) \leq r_2(G) \leq 2 \cdot \min b(G) - 1$

Proof: Consider the following steps.

- Let $v \in V$ with $b(v, G) = \min b(G) = \min a(G) = z$.
- Let $A = E_1, E_2, \cdots E_z$ be the corresponding one-way broadcast-algorithm.
- Let $B = F_1, F_2, \cdots F_z$ be the corresponding one-way accumulation-algorithm.
- Then is $F_1, F_2, \cdots F_z, E_1, E_2, \cdots E_z$ one-way gossip-algorithm.
- Note: in the two-way case holds: $F_z = E_1$.
- Note: For $L(2 \cdot n)$ we have equality.
Lemma:

For each graph $G$ with $|V| \geq 2$ we have:

- $b(G) \leq r(G) \leq 2 \cdot \min b(G)$
- $b(G) \leq r_2(G) \leq 2 \cdot \min b(G) - 1$

Proof: Consider the following steps.

- Let $v \in V$ with $b(v, G) = \min b(G) = \text{mina}(G) = z$.
- Let $A = E_1, E_2, \cdots E_z$ be the corresponding one-way broadcast-algorithm.
- Let $B = F_1, F_2, \cdots F_z$ be the corresponding one-way accumulation-algorithm.
- Then is $F_1, F_2, \cdots F_z, E_1, E_2, \cdots E_z$ one-way gossip-algorithm.
- Note: in the two-way case holds: $F_z = E_1$.
- Note: For $L(2 \cdot n)$ we have equality.
Lemma:

For each graph $G$ with $|V| \geq 2$ we have:

- $b(G) \leq r(G) \leq 2 \cdot \text{minb}(G)$
- $b(G) \leq r_2(G) \leq 2 \cdot \text{minb}(G) - 1$

Proof: Consider the following steps.

- Let $v \in V$ with $b(v, G) = \text{minb}(G) = \text{mina}(G) = z$.
- Let $A = E_1, E_2, \cdots, E_z$ be the corresponding one-way broadcast-algorithm.
- Let $B = F_1, F_2, \cdots, F_z$ be the corresponding one-way accumulation-algorithm.
- Then is $F_1, F_2, \cdots, F_z, E_1, E_2, \cdots, E_z$ one-way gossip-algorithm.
- Note: in the two-way case holds: $F_z = E_1$.
- Note: For $L(2 \cdot n)$ we have equality.
Lemma:

For each graph $G$ with $|V| \geq 2$ we have:

1. $b(G) \leq r(G) \leq 2 \cdot \text{minb}(G)$
2. $b(G) \leq r_2(G) \leq 2 \cdot \text{minb}(G) - 1$

Proof: Consider the following steps.

1. Let $v \in V$ with $b(v, G) = \text{minb}(G) = \text{mina}(G) = z$.
2. Let $A = E_1, E_2, \cdots E_z$ be the corresponding one-way broadcast-algorithm.
3. Let $B = F_1, F_2, \cdots F_z$ be the corresponding one-way accumulation-algorithm.
4. Then is $F_1, F_2, \cdots F_z, E_1, E_2, \cdots E_z$ one-way gossip-algorithm.
5. Note: in the two-way case holds: $F_z = E_1$.
6. Note: For $L(2 \cdot n)$ we have equality.
Lemma:

For each graph $G$ with $|V| \geq 2$ we have:

- $b(G) \leq r(G) \leq 2 \cdot \min b(G)$
- $b(G) \leq r_2(G) \leq 2 \cdot \min b(G) - 1$

Proof: Consider the following steps.

- Let $v \in V$ with $b(v, G) = \min b(G) = \min a(G) = z$.
- Let $A = E_1, E_2, \cdots E_z$ be the corresponding one-way broadcast-algorithm.
- Let $B = F_1, F_2, \cdots F_z$ be the corresponding one-way accumulation-algorithm.
- Then is $F_1, F_2, \cdots F_z, E_1, E_2, \cdots E_z$ one-way gossip-algorithm.

Note: in the two-way case holds: $F_z = E_1$.

Note: For $L(2 \cdot n)$ we have equality.
First Results II

Lemma:

For each graph $G$ with $|V| \geq 2$ we have:

- $b(G) \leq r(G) \leq 2 \cdot \min b(G)$
- $b(G) \leq r_2(G) \leq 2 \cdot \min b(G) - 1$

Proof: Consider the following steps.

- Let $v \in V$ with $b(v, G) = \min b(G) = \min a(G) = z$.
- Let $A = E_1, E_2, \cdots, E_z$ be the corresponding one-way broadcast-algorithm.
- Let $B = F_1, F_2, \cdots, F_z$ be the corresponding one-way accumulation-algorithm.
- Then is $F_1, F_2, \cdots, F_z, E_1, E_2, \cdots, E_z$ one-way gossip-algorithm.
- Note: in the two-way case holds: $F_z = E_1$.
- Note: For $L(2 \cdot n)$ we have equality.
Lemma:
For each even $n$ with $n \geq 8$ exists a Graph $G$ with $n$ nodes and $b(G) = r(G)$

Proof (for $n = 8$):

- Both broadcasts together are a gossip-algorithm.
Lemma:

For each even \( n \) with \( n \geq 8 \) exists a Graph \( G \) with \( n \) nodes and

\[ b(G) = r(G) \]

Proof (for \( n = 8 \)):

Both broadcasts together are a gossip-algorithm.
Lemma:
For each even $n$ with $n \geq 8$ exists a Graph $G$ with $n$ nodes and
\[ b(G) = r(G) \]

Proof (for $n = 8$):

Both broadcasts together are a gossip-algorithm.
Lemma:
For each even $n$ with $n \geq 8$ exists a Graph $G$ with $n$ nodes and $b(G) = r(G)$.

Proof (for $n = 8$):

Both broadcasts together are a gossip-algorithm.
First Results III

Lemma:
For each even \( n \) with \( n \geq 8 \) exists a Graph \( G \) with \( n \) nodes and
\[
b(G) = r(G)
\]

Proof (for \( n = 8 \)): 

Both broadcasts together are a gossip-algorithm.
Lemma:
For each even $n$ with $n \geq 8$ exists a Graph $G$ with $n$ nodes and
\[ b(G) = r(G) \]

Proof (for $n = 8$):

Both broadcasts together are a gossip-algorithm.
Lemma:
For each even $n$ with $n \geq 8$ exists a Graph $G$ with $n$ nodes and
\[ b(G) = r(G) \]

Proof (for $n = 8$):

Both broadcasts together are a gossip-algorithm.
First Results III

Lemma:
For each even $n$ with $n \geq 8$ exists a Graph $G$ with $n$ nodes and

$$b(G) = r(G)$$

Proof (for $n = 8$):

Both broadcasts together are a gossip-algorithm.
Lemma:
For each even $n$ with $n \geq 8$ exists a Graph $G$ with $n$ nodes and
\[ b(G) = r(G) \]

Proof (for $n = 8$):
Lemma:
For each even $n$ with $n \geq 8$ exists a Graph $G$ with $n$ nodes and
$$b(G) = r(G)$$

Proof (for $n = 8$):
Lemma:
For each even $n$ with $n \geq 8$ exists a Graph $G$ with $n$ nodes and

$\bullet$ $b(G) = r(G)$

Proof (for $n = 8$):

Both broadcasts together are a gossip-algorithm.
Lemma:
For each even $n$ with $n \geq 8$ exists a Graph $G$ with $n$ nodes and $b(G) = r(G)$

Proof (for $n = 8$):
Lemma:

For each even $n$ with $n \geq 8$ exists a Graph $G$ with $n$ nodes and $b(G) = r(G)$

Proof (for $n = 8$):

Both broadcasts together are a gossip-algorithm.
Lemma:

For each even \( n \) with \( n \geq 8 \) exists a Graph \( G \) with \( n \) nodes and

\[ b(G) = r(G) \]

Proof (for \( n = 8 \)):
Lemma:

For each even $n$ with $n \geq 8$ exists a Graph $G$ with $n$ nodes and $b(G) = r(G)$.

Proof (for $n = 8$):

Both broadcasts together are a gossip algorithm.
First Results III

Lemma:
For each even $n$ with $n \geq 8$ exists a Graph $G$ with $n$ nodes and

$$b(G) = r(G)$$

Proof (for $n = 8$):

Both broadcasts together are a gossip-algorithm.
First Results IV

- \( \text{rad}(G) \leq \text{minb}(G) \).
- \( \text{rad}(G) \leq \text{diam}(G) \leq b(G) \).
- Let \( G = (V, E) \) and \( H = (V, F) \) with \( F \subseteq E \). Then we have:
  - \( b(G) \leq b(H) \).
  - \( \text{minb}(G) \leq \text{minb}(H) \).
  - \( r(G) \leq r(H) \).
  - \( r_2(G) \leq r_2(H) \).

- \( \text{minb}(G) \leq (\text{deg}(G) - 1) \cdot \text{rad}(G) + 1 \).
- \( b(G) \leq (\text{deg}(G) - 1) \cdot \text{diam}(G) + 1 \).
- \( b(G) \leq \text{deg}(G) \cdot \text{rad}(G) \).
- \( r(G) \leq 2(\text{deg}(G) - 1) \cdot \text{rad}(G) + 2 \)
- \( r_2(G) \leq 2(\text{deg}(G) - 1) \cdot \text{rad}(G) + 1 \)
First Results IV

- \( \text{rad}(G) \leq \text{minb}(G) \).
- \( \text{rad}(G) \leq \text{diam}(G) \leq b(G) \).

Let \( G = (V, E) \) and \( H = (V, F) \) with \( F \subseteq E \). Then we have:

- \( b(G) \leq b(H) \).
- \( \text{minb}(G) \leq \text{minb}(H) \).
- \( r(G) \leq r(H) \).
- \( r_2(G) \leq r_2(H) \).

- \( \text{minb}(G) \leq (\text{deg}(G) - 1) \cdot \text{rad}(G) + 1 \).
- \( b(G) \leq (\text{deg}(G) - 1) \cdot \text{diam}(G) + 1 \).
- \( b(G) \leq \text{deg}(G) \cdot \text{rad}(G) \).
- \( r(G) \leq 2(\text{deg}(G) - 1) \cdot \text{rad}(G) + 2 \).
- \( r_2(G) \leq 2(\text{deg}(G) - 1) \cdot \text{rad}(G) + 1 \)
First Results IV

- \( \text{rad}(G) \leq \text{minb}(G) \).

- \( \text{rad}(G) \leq \text{diam}(G) \leq b(G) \).

- Let \( G = (V, E) \) and \( H = (V, F) \) with \( F \subseteq E \). Then we have:
  - \( b(G) \leq b(H) \).
  - \( \text{minb}(G) \leq \text{minb}(H) \).
  - \( r(G) \leq r(H) \).
  - \( r_2(G) \leq r_2(H) \).

- \( \text{minb}(G) \leq (\text{deg}(G) - 1) \cdot \text{rad}(G) + 1 \).

- \( b(G) \leq (\text{deg}(G) - 1) \cdot \text{diam}(G) + 1 \).

- \( b(G) \leq \text{deg}(G) \cdot \text{rad}(G) \).

- \( r(G) \leq 2(\text{deg}(G) - 1) \cdot \text{rad}(G) + 2 \).

- \( r_2(G) \leq 2(\text{deg}(G) - 1) \cdot \text{rad}(G) + 1 \).
First Results IV

- $\text{rad}(G) \leq \text{minb}(G)$.
- $\text{rad}(G) \leq \text{diam}(G) \leq b(G)$.
- Let $G = (V, E)$ and $H = (V, F)$ with $F \subseteq E$. Then we have:
  - $b(G) \leq b(H)$.
  - $\text{minb}(G) \leq \text{minb}(H)$.
  - $r(G) \leq r(H)$.
  - $r_2(G) \leq r_2(H)$.

\[
\begin{align*}
\text{diam}(G) &= \max\{\text{dist}(u, v) \mid u, v \in V\} \\
\text{rad}(v, G) &= \max\{\text{dist}(v, x) \mid x \in V\} \\
\text{rad}(G) &= \min\{\text{rad}(v, G) \mid v \in V\}
\end{align*}
\]
First Results IV

- \( \text{rad}(G) \leq \text{minb}(G) \).
- \( \text{rad}(G) \leq \text{diam}(G) \leq b(G) \).

Let \( G = (V, E) \) and \( H = (V, F) \) with \( F \subseteq E \). Then we have:

- \( b(G) \leq b(H) \).
- \( \text{minb}(G) \leq \text{minb}(H) \).
- \( r(G) \leq r(H) \).
- \( r_2(G) \leq r_2(H) \).

- \( \text{minb}(G) \leq (\text{deg}(G) - 1) \cdot \text{rad}(G) + 1 \).
- \( b(G) \leq (\text{deg}(G) - 1) \cdot \text{diam}(G) + 1 \).
- \( b(G) \leq \text{deg}(G) \cdot \text{rad}(G) \).
- \( r(G) \leq 2(\text{deg}(G) - 1) \cdot \text{rad}(G) + 2 \)
- \( r_2(G) \leq 2(\text{deg}(G) - 1) \cdot \text{rad}(G) + 1 \)
First Results IV

- \( \text{rad}(G) \leq \text{minb}(G) \).
- \( \text{rad}(G) \leq \text{diam}(G) \leq b(G) \).

Let \( G = (V, E) \) and \( H = (V, F) \) with \( F \subseteq E \). Then we have:
- \( b(G) \leq b(H) \).
- \( \text{minb}(G) \leq \text{minb}(H) \).
- \( r(G) \leq r(H) \).
- \( r_2(G) \leq r_2(H) \).

\[
\begin{align*}
\text{diam}(G) & = \max\{\text{dist}(u, v) \mid u, v \in V\} \\
\text{rad}(v, G) & = \max\{\text{dist}(v, x) \mid x \in V\} \\
\text{rad}(G) & = \min\{\text{rad}(v, G) \mid v \in V\}
\end{align*}
\]
First Results IV

- $\text{rad}(G) \leq \text{minb}(G)$.  
- $\text{rad}(G) \leq \text{diam}(G) \leq b(G)$.  

Let $G = (V, E)$ and $H = (V, F)$ with $F \subseteq E$. Then we have:  
  - $b(G) \leq b(H)$.  
  - $\text{minb}(G) \leq \text{minb}(H)$.  
  - $\text{r}(G) \leq \text{r}(H)$.  
  - $r_2(G) \leq r_2(H)$.  

- $\text{minb}(G) \leq (\text{deg}(G) - 1) \cdot \text{rad}(G) + 1$.  
- $b(G) \leq (\text{deg}(G) - 1) \cdot \text{diam}(G) + 1$.  
- $b(G) \leq \text{deg}(G) \cdot \text{rad}(G)$.  
- $\text{r}(G) \leq 2(\text{deg}(G) - 1) \cdot \text{rad}(G) + 2$  
- $r_2(G) \leq 2(\text{deg}(G) - 1) \cdot \text{rad}(G) + 1$
First Results IV

- $\text{rad}(G) \leq \text{minb}(G)$.
- $\text{rad}(G) \leq \text{diam}(G) \leq b(G)$.
- Let $G = (V, E)$ and $H = (V, F)$ with $F \subseteq E$. Then we have:
  - $b(G) \leq b(H)$.
  - $\text{minb}(G) \leq \text{minb}(H)$.
  - $r(G) \leq r(H)$.
  - $r_2(G) \leq r_2(H)$.
- $\text{minb}(G) \leq (\text{deg}(G) - 1) \cdot \text{rad}(G) + 1$.
- $b(G) \leq (\text{deg}(G) - 1) \cdot \text{diam}(G) + 1$.
- $b(G) \leq \text{deg}(G) \cdot \text{rad}(G)$.
- $r(G) \leq 2(\text{deg}(G) - 1) \cdot \text{rad}(G) + 2$
- $r_2(G) \leq 2(\text{deg}(G) - 1) \cdot \text{rad}(G) + 1$

\[
\begin{align*}
\text{diam}(G) &= \max \{\text{dist}(u, v) \mid u, v \in V\} \\
\text{rad}(v, G) &= \max \{\text{dist}(v, x) \mid x \in V\} \\
\text{rad}(G) &= \min \{\text{rad}(v, G) \mid v \in V\}
\end{align*}
\]
First Results IV

- \( \text{rad}(G) \leq \text{minb}(G) \).
- \( \text{rad}(G) \leq \text{diam}(G) \leq b(G) \).

Let \( G = (V, E) \) and \( H = (V, F) \) with \( F \subseteq E \). Then we have:
  - \( b(G) \leq b(H) \).
  - \( \text{minb}(G) \leq \text{minb}(H) \).
  - \( r(G) \leq r(H) \).
  - \( r_2(G) \leq r_2(H) \).

- \( \text{minb}(G) \leq (\text{deg}(G) - 1) \cdot \text{rad}(G) + 1 \).
- \( b(G) \leq (\text{deg}(G) - 1) \cdot \text{diam}(G) + 1 \).
- \( b(G) \leq \text{deg}(G) \cdot \text{rad}(G) \).
- \( r(G) \leq 2(\text{deg}(G) - 1) \cdot \text{rad}(G) + 2 \).
- \( r_2(G) \leq 2(\text{deg}(G) - 1) \cdot \text{rad}(G) + 1 \).
First Results IV

- $\text{rad}(G) \leq \text{minb}(G)$.
- $\text{rad}(G) \leq \text{diam}(G) \leq b(G)$.
- Let $G = (V, E)$ and $H = (V, F)$ with $F \subset E$. Then we have:
  - $b(G) \leq b(H)$.
  - $\text{minb}(G) \leq \text{minb}(H)$.
  - $r(G) \leq r(H)$.
  - $r_2(G) \leq r_2(H)$.
- $\text{minb}(G) \leq (\text{deg}(G) - 1) \cdot \text{rad}(G) + 1$.
- $b(G) \leq (\text{deg}(G) - 1) \cdot \text{diam}(G) + 1$.
- $b(G) \leq \text{deg}(G) \cdot \text{rad}(G)$.
- $r(G) \leq 2(\text{deg}(G) - 1) \cdot \text{rad}(G) + 2$
- $r_2(G) \leq 2(\text{deg}(G) - 1) \cdot \text{rad}(G) + 1$.
First Results IV

- \( \text{rad}(G) \leq \text{minb}(G) \).
- \( \text{rad}(G) \leq \text{diam}(G) \leq b(G) \).
- Let \( G = (V, E) \) and \( H = (V, F) \) with \( F \subseteq E \). Then we have:
  - \( b(G) \leq b(H) \).
  - \( \text{minb}(G) \leq \text{minb}(H) \).
  - \( r(G) \leq r(H) \).
  - \( r_2(G) \leq r_2(H) \).
- \( \text{minb}(G) \leq (\text{deg}(G) - 1) \cdot \text{rad}(G) + 1 \).
- \( b(G) \leq (\text{deg}(G) - 1) \cdot \text{diam}(G) + 1 \).
- \( b(G) \leq \text{deg}(G) \cdot \text{rad}(G) \).
- \( r(G) \leq 2(\text{deg}(G) - 1) \cdot \text{rad}(G) + 2 \).
- \( r_2(G) \leq 2(\text{deg}(G) - 1) \cdot \text{rad}(G) + 1 \)
First Results IV

- \( \text{rad}(G) \leq \text{minb}(G) \).
- \( \text{rad}(G) \leq \text{diam}(G) \leq b(G) \).

Let \( G = (V, E) \) and \( H = (V, F) \) with \( F \subseteq E \). Then we have:
  - \( b(G) \leq b(H) \).
  - \( \text{minb}(G) \leq \text{minb}(H) \).
  - \( r(G) \leq r(H) \).
  - \( r_2(G) \leq r_2(H) \).

- \( \text{minb}(G) \leq (\deg(G) - 1) \cdot \text{rad}(G) + 1 \).
- \( b(G) \leq (\deg(G) - 1) \cdot \text{diam}(G) + 1 \).
- \( b(G) \leq \deg(G) \cdot \text{rad}(G) \).
- \( r(G) \leq 2(\deg(G) - 1) \cdot \text{rad}(G) + 2 \).
- \( r_2(G) \leq 2(\deg(G) - 1) \cdot \text{rad}(G) + 1 \).
Lower Bound

Lemma

Let $G = (V, E)$ be a graph with $n$ nodes. Then we have:

- $b(G) \geq \min b(G) \geq \lceil \log n \rceil$

Proof:

- Let $A(t)$ be the number of informed nodes after $t$ rounds.
- $A(0) = 1$
- $A(t + 1) \leq 2 \cdot A(t)$
- $A(t) \leq 2^t$
- At the end $2^t \geq n$ must hold.
Lower Bound

Lemma

Let $G = (V, E)$ be a graph with $n$ nodes. Then we have:

- $b(G) \geq \min b(G) \geq \lceil \log n \rceil$

Proof:

- Let $A(t)$ be the number of informed nodes after $t$ rounds.
- $A(0) = 1$
- $A(t + 1) \leq 2 \cdot A(t)$
- $A(t) \leq 2^t$
- At the end $2^t \geq n$ must hold.
Lower Bound

Lemma

Let $G = (V, E)$ be a graph with $n$ nodes. Then we have:

- $b(G) \geq \min b(G) \geq \lceil \log n \rceil$

Proof:

- Let $A(t)$ be the number of informed nodes after $t$ rounds.
- $A(0) = 1$
- $A(t + 1) \leq 2 \cdot A(t)$
- $A(t) \leq 2^t$
- At the end $2^t \geq n$ must hold.
Lower Bound

Lemma

Let $G = (V, E)$ be a graph with $n$ nodes. Then we have:

- $b(G) \geq \text{minb}(G) \geq \lceil \log n \rceil$

Proof:

- Let $A(t)$ be the number of informed nodes after $t$ rounds.
- $A(0) = 1$
- $A(t + 1) \leq 2 \cdot A(t)$
- $A(t) \leq 2^t$
- At the end $2^t \geq n$ must hold.
Lemma
Let $G = (V, E)$ be a graph with $n$ nodes. Then we have:

- $b(G) \geq \min b(G) \geq \lceil \log n \rceil$

Proof:
- Let $A(t)$ be the number of informed nodes after $t$ rounds.
- $A(0) = 1$
- $A(t + 1) \leq 2 \cdot A(t)$
- $A(t) \leq 2^t$
- At the end $2^t \geq n$ must hold.
Lemma

Let $G = (V, E)$ be a graph with $n$ nodes. Then we have:

- $b(G) \geq \min b(G) \geq \lceil \log n \rceil$

Proof:

- Let $A(t)$ be the number of informed nodes after $t$ rounds.
- $A(0) = 1$
- $A(t + 1) \leq 2 \cdot A(t)$
- $A(t) \leq 2^t$
- At the end $2^t \geq n$ must hold.
Optimal Broadcast-Tree

Each informed node has to send in each round the information to a non-informed node:

A tree $T_i$ is a broadcast-tree, iff

- the root of $T_i$ has $i$ successors $v_0, v_1, \cdots, v_{i-1}$ and
- $v_j$ is the root of a $T_j$. 
Optimal Broadcast-Tree

Each informed node has to send in each round the information to a non-informed node:

A tree $T_i$ is a broadcast-tree, iff

- the root of $T_i$ has $i$ successors $v_0, v_1, \cdots, v_{i-1}$ and
- $v_j$ is the root of a $T_j$. 

Optimal Broadcast-Tree

Each informed node has to send in each round the information to a non-informed node:

A tree $T_i$ is a broadcast-tree, iff
- the root of $T_i$ has $i$ successors $v_0, v_1, \cdots, v_{i-1}$ and
- $v_j$ is the root of a $T_j$. 

A tree $T_i$ is a broadcast-tree, iff
Optimal Broadcast-Tree

Each informed node has to send in each round the information to a non-informed node:

A tree $T_i$ is a broadcast-tree, iff
- the root of $T_i$ has $i$ successors $v_0, v_1, \cdots, v_{i-1}$ and
- $v_j$ is the root of a $T_j$. 
Optimal Broadcast-Tree

Each informed node has to send in each round the information to a non-informed node:

A tree $T_i$ is a broadcast-tree, iff

- the root of $T_i$ has $i$ successors $v_0, v_1, \cdots, v_{i-1}$ and
- $v_j$ is the root of a $T_j$. 

![Diagram of a broadcast-tree with nodes labeled 0 to 9, illustrating the optimal broadcast](image-url)
Optimal Broadcast-Tree

Each informed node has to send in each round the information to a non-informed node:

A tree $T_i$ is a broadcast-tree, iff

- the root of $T_i$ has $i$ successors $v_0, v_1, \ldots, v_{i-1}$ and
- $v_j$ is the root of a $T_j$. 

```plaintext
A tree $T_i$ is a broadcast-tree, iff

- the root of $T_i$ has $i$ successors $v_0, v_1, \ldots, v_{i-1}$ and
- $v_j$ is the root of a $T_j$.
```
Optimal Broadcast-Tree

Each informed node has to send in each round the information to a non-informed node:

A tree $T_i$ is a broadcast-tree, iff
- the root of $T_i$ has $i$ successors $v_0, v_1, \cdots, v_{i-1}$ and
- $v_j$ is the root of a $T_j$. 
Optimal Broadcast-Tree

Each informed node has to send in each round the information to a non-informed node:

- A tree $T_i$ is a broadcast-tree, iff
  - the root of $T_i$ has $i$ successors $v_0, v_1, \cdots, v_{i-1}$ and
  - $v_j$ is the root of a $T_j$. 
Optimal Broadcast-Tree

Each informed node has to send in each round the information to a non-informed node:

A tree \( T_i \) is a broadcast-tree, iff

1. the root of \( T_i \) has \( i \) successors \( v_0, v_1, \ldots, v_{i-1} \) and
2. \( v_j \) is the root of a \( T_j \).
Optimal Broadcast-Tree

Each informed node has to send in each round the information to a non-informed node:

A tree $T_i$ is a broadcast-tree, iff
- the root of $T_i$ has $i$ successors $v_0, v_1, \ldots, v_{i-1}$ and
- $v_j$ is the root of a $T_j$. 
Optimal Broadcast-Tree

Each informed node has to send in each round the information to a non-informed node:

- A tree $T_i$ is a broadcast-tree, iff
  - the root of $T_i$ has $i$ successors $v_0, v_1, \ldots, v_{i-1}$ and
  - $v_j$ is the root of a $T_j$. 
Each informed node has to send in each round the information to a non-informed node:

A tree $T_i$ is a broadcast-tree, iff
- the root of $T_i$ has $i$ successors $v_0, v_1, \ldots, v_{i-1}$ and
- $v_j$ is the root of a $T_j$.  


Each informed node has to send in each round the information to a non-informed node:

A tree $T_i$ is a broadcast-tree, iff

- the root of $T_i$ has $i$ successors $v_0, v_1, \ldots, v_{i-1}$ and
- $v_j$ is the root of a $T_j$. 
First Results

Lemma

We have:

- \( \min b(K(n)) = b(K(n)) = \lceil \log n \rceil \) and
- \( \min b(HQ(m)) = b(HQ(m)) = m. \)

Proof \((K(n))\):

for \( t = 1 \) to \( \lceil \log n \rceil \) do

for all \( i \in \{0, 1, \ldots, 2^{t-1} - 1\} \) do in parallel

if \( i + 2^{t-1} \leq n \) then

\( i \) sends to \( i + 2^{t-1} \)

Proof \((HQ(m))\):

for \( i = 1 \) to \( m \) do

for all \( a_1, a_2, \ldots, a_{i-1} \in \{0, 1\} \) do in parallel

\( a_1a_2\ldots a_{i-1}00\ldots0 \) sends to \( a_1a_2\ldots a_{i-1}10\ldots0 \)
Lemma

We have:

- \( \min b(K(n)) = b(K(n)) = \lceil \log n \rceil \) and
- \( \min b(HQ(m)) = b(HQ(m)) = m. \)

Proof \( (K(n)):\)

\[
\text{for } t = 1 \text{ to } \lceil \log n \rceil \text{ do } \\
\quad \text{for all } i \in \{0, 1, \ldots, 2^{t-1} - 1\} \text{ do in parallel } \\
\quad \quad \text{if } i + 2^{t-1} \leq n \text{ then } \\
\quad \quad \quad i \text{ sends to } i + 2^{t-1}
\]

Proof \( (HQ(m)):\)

\[
\text{for } i = 1 \text{ to } m \text{ do } \\
\quad \text{for all } a_1, a_2, \ldots, a_{i-1} \in \{0, 1\} \text{ do in parallel } \\
\quad \quad a_1a_2\cdots a_{i-1}00\cdots0 \text{ sends to } a_1a_2\cdots a_{i-1}10\cdots0
\]
First Results

Lemma

We have:

- $\min b(K(n)) = b(K(n)) = \lceil \log n \rceil$ and $\min b(HQ(m)) = b(HQ(m)) = m$.

Proof ($K(n)$):

for $t = 1$ to $\lceil \log n \rceil$ do

for all $i \in \{0, 1, \ldots, 2^{t-1} - 1\}$ do in parallel

if $i + 2^{t-1} \leq n$ then

$i$ sends to $i + 2^{t-1}$

Proof ($HQ(m)$):

for $i = 1$ to $m$ do

for all $a_1, a_2, \ldots, a_{i-1} \in \{0, 1\}$ do in parallel

$a_1a_2\cdots a_{i-1}00\cdots 0$ sends to $a_1a_2\cdots a_{i-1}10\cdots 0$
First Results II

**Lemma**

For all \( k, m \geq 2 \) we have: \( \min b(T_k(m)) = k \cdot m \).

Idea of proof:

- \( b(\varepsilon, T_k(m)) = k \cdot m \).
- \( b(\varepsilon, T_k(m)) \leq b(\nu, T_k(m)) \).
- Note that \( \nu \) has to inform \( \varepsilon \)
- and \( \varepsilon \) has to inform the other successors.
Lemma

For all $k, m \geq 2$ we have: $\min b(T_k(m)) = k \cdot m$.

Idea of proof:

1. $b(\varepsilon, T_k(m)) = k \cdot m$.
2. $b(\varepsilon, T_k(m)) \leq b(\nu, T_k(m))$.
3. Note that $\nu$ has to inform $\varepsilon$.
4. and $\varepsilon$ has to inform the other successors.
First Results II

**Lemma**

For all $k, m \geq 2$ we have: $\minb(T_k(m)) = k \cdot m$.

**Idea of proof:**

- $b(\varepsilon, T_k(m)) = k \cdot m$.
- $b(\varepsilon, T_k(m)) \leq b(\nu, T_k(m))$.
- Note that $\nu$ has to inform $\varepsilon$.
- And $\varepsilon$ has to inform the other successors.
First Results II

Lemma

For all \( k, m \geq 2 \) we have: \( \min_b(T_k(m)) = k \cdot m \).

Idea of proof:

- \( b(\varepsilon, T_k(m)) = k \cdot m \).
- \( b(\varepsilon, T_k(m)) \leq b(\nu, T_k(m)) \).
- Note that \( \nu \) has to inform \( \varepsilon \)
- and \( \varepsilon \) has to inform the other successors.
Lemma

For all \( k, m \geq 2 \) we have: \( \min b(T_k(m)) = k \cdot m \).

Idea of proof:

\[ b(\varepsilon, T_k(m)) = k \cdot m. \]

\[ b(\varepsilon, T_k(m)) \leq b(\nu, T_k(m)). \]

Note that \( \nu \) has to inform \( \varepsilon \)

and \( \varepsilon \) has to inform the other successors.
First Results

Lemma

We have:

- $b(\text{CCC}(k)) \leq 5k + O(1)$
- $b(\text{BF}(k)) \leq 4.5k + O(1)$
- $b(\text{SE}(k)) \leq 4k + O(1)$
- $b(\text{DB}(k)) \leq 3k + O(1)$

Proof: Use the following statements:

- $b(G) \leq (\text{deg}(G) - 1) \cdot \text{diam}(G) + 1$.
- $b(G) \leq \text{deg}(G) \cdot \text{rad}(G)$. 

First Results

Lemma

We have:

- \( b(\text{CCC}(k)) \leq 5k + O(1) \)
- \( b(\text{BF}(k)) \leq 4.5k + O(1) \)
- \( b(\text{SE}(k)) \leq 4k + O(1) \)
- \( b(\text{DB}(k)) \leq 3k + O(1) \)

Proof: Use the following statements:

- \( b(G) \leq (\text{deg}(G) - 1) \cdot \text{diam}(G) + 1 \).
- \( b(G) \leq \text{deg}(G) \cdot \text{rad}(G) \).
First Results

Lemma

We have:

- \( b(\text{CCC}(k)) \leq 5k + O(1) \)
- \( b(\text{BF}(k)) \leq 4.5k + O(1) \)
- \( b(\text{SE}(k)) \leq 4k + O(1) \)
- \( b(\text{DB}(k)) \leq 3k + O(1) \)

Proof: Use the following statements:

- \( b(G) \leq (\text{deg}(G) - 1) \cdot \text{diam}(G) + 1. \)
- \( b(G) \leq \text{deg}(G) \cdot \text{rad}(G). \)
First Results

Lemma

We have:

- \( b(\text{CCC}(k)) \leq 5k + O(1) \)
- \( b(\text{BF}(k)) \leq 4.5k + O(1) \)
- \( b(\text{SE}(k)) \leq 4k + O(1) \)
- \( b(\text{DB}(k)) \leq 3k + O(1) \)

Proof: Use the following statements:

- \( b(G) \leq (\text{deg}(G) - 1) \cdot \text{diam}(G) + 1. \)
- \( b(G) \leq \text{deg}(G) \cdot \text{rad}(G). \)
First Results

Lemma

We have:

- \( b(\text{CCC}(k)) \leq 5k + O(1) \)
- \( b(\text{BF}(k)) \leq 4.5k + O(1) \)
- \( b(\text{SE}(k)) \leq 4k + O(1) \)
- \( b(\text{DB}(k)) \leq 3k + O(1) \)

Proof: Use the following statements:

- \( b(G) \leq (\deg(G) - 1) \cdot \diam(G) + 1. \)
- \( b(G) \leq \deg(G) \cdot \rad(G). \)
First Results

Lemma

We have:

- \( b(\text{CCC}(k)) \leq 5k + O(1) \)
- \( b(\text{BF}(k)) \leq 4.5k + O(1) \)
- \( b(\text{SE}(k)) \leq 4k + O(1) \)
- \( b(\text{DB}(k)) \leq 3k + O(1) \)

Proof: Use the following statements:

- \( b(G) \leq (\deg(G) - 1) \cdot \diam(G) + 1. \)
- \( b(G) \leq \deg(G) \cdot \rad(G). \)
Theorem:

We have: \( \lceil \frac{5k}{2} \rceil - 2 \leq \min_b(\text{CCC}(k)) = b(\text{CCC}(k)) \leq \lceil \frac{5k}{2} \rceil - 1 \).

- The following parts are proven:
  - \( \min_b(\text{CCC}(k)) \geq \lceil \frac{5k}{2} \rceil - 2 \)
  - Algorithm for \( \lceil \frac{5k}{2} \rceil - 1 \) will be presented.
Theorem:

We have: \( \left\lceil \frac{5k}{2} \right\rceil - 2 \leq \min b(\text{CCC}(k)) = b(\text{CCC}(k)) \leq \left\lceil \frac{5k}{2} \right\rceil - 1. \)

- The following parts are proven:
  - \( \min b(\text{CCC}(k)) \geq \left\lceil \frac{5k}{2} \right\rceil - 2 \)
  - Algorithm for \( \left\lceil \frac{5k}{2} \right\rceil - 1 \) will be presented.
Theorem:

We have: \( \left\lceil \frac{5k}{2} \right\rceil - 2 \leq \min_b(\text{CCC}(k)) = b(\text{CCC}(k)) \leq \left\lceil \frac{5k}{2} \right\rceil - 1 \).

- The following parts are proven:
  - \( \min_b(\text{CCC}(k)) \geq \left\lceil \frac{5k}{2} \right\rceil - 2 \)
  - Algorithm for \( \left\lceil \frac{5k}{2} \right\rceil - 1 \) will be presented.
CCC, Proof \( \text{minb}(\text{CCC}(k)) \geq \lfloor 5 \cdot k/2 \rfloor - 2 \)

- \( \text{diam}(\text{CCC}(k)) = \lfloor 5/2 \cdot k \rfloor - 2 \)
- The statement holds for even \( k \).
- Let \( k \) be odd.
- Let \((0,00\cdots0)\) be the origin of the message.
- The nodes \((\lfloor k/2 \rfloor, 11\cdots1)\) and \((\lfloor k/2 \rfloor + 1, 11\cdots1)\) are both in distance \((\lfloor 5 \cdot k/2 \rfloor - 2)\).
- Thus we need one round more than the diameter.
- The statement hold, because the CCC is node-symmetric.
CCC, Proof \( \text{minb}(\text{CCC}(k)) \geq \lceil 5 \cdot k/2 \rceil - 2 \)

- \( \text{diam}(\text{CCC}(k)) = \lceil 5/2 \cdot k \rceil - 2 \)
- The statement holds for even \( k \).
- Let \( k \) be odd.
- Let \((0,00 \cdots 0)\) be the origin of the message.
- The nodes \( (\lfloor k/2 \rfloor, 11 \cdots 1) \) and \( (\lfloor k/2 \rfloor + 1, 11 \cdots 1) \) are both in distance \( \lfloor 5 \cdot k/2 \rfloor - 2 \).
- Thus we need one round more then the diameter.
- The statement hold, because the CCC is node-symmetric.
CCC, Proof $\text{minb}(\text{CCC}(k)) \geq \lceil 5 \cdot k/2 \rceil - 2$

- $\text{diam}(\text{CCC}(k)) = \lceil 5/2 \cdot k \rceil - 2$
- The statement holds for even $k$.
- Let $k$ be odd.
- Let $(0,00\cdots0)$ be the origin of the message.
- The nodes $([k/2], 11\cdots1)$ and $([k/2]+1, 11\cdots1)$ are both in distance $(\lceil 5 \cdot k/2 \rceil - 2)$.
- Thus we need one round more than the diameter.
- The statement hold, because the CCC is node-symmetric.
**CCC, Proof**

\[ \text{minb}(\text{CCC}(k)) \geq \lceil 5 \cdot k/2 \rceil - 2 \]

- \( \text{diam}(\text{CCC}(k)) = \lceil 5/2 \cdot k \rceil - 2 \)
- The statement holds for even \( k \).
- Let \( k \) be odd.
- Let \((0,00\cdots0)\) be the origin of the message.
- The nodes \((\lfloor k/2 \rfloor, 11\cdots1)\) and \((\lfloor k/2 \rfloor + 1, 11\cdots1)\) are both in distance \((\lceil 5 \cdot k/2 \rceil - 2)\).
- Thus we need one round more than the diameter.
- The statement holds, because the CCC is node-symmetric.
CCC, Proof $\text{minb}(\text{CCC}(k)) \geq \lceil 5 \cdot k/2 \rceil - 2$

- $\text{diam}(\text{CCC}(k)) = \lceil 5/2 \cdot k \rceil - 2$
- The statement holds for even $k$.
- Let $k$ be odd.
- Let $(0, 00 \cdots 0)$ be the origin of the message.
- The nodes $(\lfloor k/2 \rfloor, 11 \cdots 1)$ and $(\lfloor k/2 \rfloor + 1, 11 \cdots 1)$ are both in distance $(\lceil 5 \cdot k/2 \rceil - 2)$.
- Thus we need one round more then the diameter.
- The statement hold, because the CCC is node-symmetric.
\[ \text{CCC, Proof } \min \text{b}(\text{CCC}(k)) \geq \left\lfloor \frac{5 \cdot k}{2} \right\rfloor - 2 \]

- \( \text{diam}(\text{CCC}(k)) = \left\lfloor \frac{5}{2} \cdot k \right\rfloor - 2 \)
- The statement holds for even \( k \).
- Let \( k \) be odd.
- Let \( (0,00 \cdots 0) \) be the origin of the message.
- The nodes \( (\left\lfloor k/2 \right\rfloor, 11 \cdots 1) \) and \( (\left\lfloor k/2 \right\rfloor + 1, 11 \cdots 1) \) are both in distance \( \left\lfloor 5 \cdot k/2 \right\rfloor - 2 \).
- Thus we need one round more than the diameter.
- The statement holds, because the CCC is node-symmetric.
CCC, Proof $\minb(\text{CCC}(k)) \geq \lceil 5 \cdot k/2 \rceil - 2$

- $\text{diam}(\text{CCC}(k)) = \lceil 5/2 \cdot k \rceil - 2$
- The statement holds for even $k$.
- Let $k$ be odd.
- Let $(0,00\cdots 0)$ be the origin of the message.
- The nodes $(\lceil k/2 \rceil, 11\cdots 1)$ and $(\lceil k/2 \rceil + 1, 11\cdots 1)$ are both in distance $(\lceil 5 \cdot k/2 \rceil - 2)$.
- Thus we need one round more then the diameter.
- The statement hold, because the CCC is node-symetric.
Algorithm BROADCAST-CCC\(_k\)
(0, 00...0) sends to (0, 10...0);
for \(i = 0\) to \(k - 1\) do begin
    for all \(a_0, \ldots, a_{i-1} \in \{0, 1\}\) do in parallel
        \((i - 1, a_0 \ldots a_{i-1}00 \ldots 0)\) sends to \((i, a_0 \ldots a_{i-1}00 \ldots 0)\);
    for all \(a_0, \ldots, a_{i-1} \in \{0, 1\}\) do in parallel
        \((i, a_0 \ldots a_{i-1}00 \ldots 0)\) sends to \((i, a_0 \ldots a_{i-1}10 \ldots 0)\);
end;
for all \(\alpha \in \{0, 1\}^k\) do in parallel
    Broadcast on cycle \(C_\alpha(k)\) starting from \((k - 1, \alpha)\);
Theorem:
We have: \( \min b(\text{CCC}(k)) = b(\text{CCC}(k)) \leq \lceil 5 \cdot k / 2 \rceil - 2. \)

Idea of proof: Change the first phase and send in both directions.
Theorem:

We have: \( \min b(\text{CCC}(k)) = b(\text{CCC}(k)) \leq \lceil 5 \cdot k/2 \rceil - 2. \)

Idea of proof: Change the first phase and send in both directions.
Theorem:

We have: \( \min b(SE(k)) = b(SE(k)) = 2 \cdot k - 1 \)

Proof:

- The diameter provides the lower bound.
- Note \( SE(k) \) is not node-symmetric.
- We have to provide an algorithm for any node \( v \).
- Algorithm has to be without conflicts.
- And we do now show it here in detail.
Theorem:

We have: \( \minb(SE(k)) = b(SE(k)) = 2 \cdot k - 1 \)

Proof:

- The diameter provides the lower bound.
- Note \( SE(k) \) is not node-symmetric.
- We have to provide an algorithm for any node \( v \).
- Algorithm has to be without conflicts.
- And we do now show it here in detail.
Theorem:

We have: \( \min b(SE(k)) = b(SE(k)) = 2 \cdot k - 1 \)

Proof:

- The diameter provides the lower bound.
- Note \( SE(k) \) is not node-symmetric.
- We have to provide an algorithm for any node \( v \).
- Algorithm has to be without conflicts.
- And we do now show it here in detail.
Theorem:

We have: \( \min b(SE(k)) = b(SE(k)) = 2 \cdot k - 1 \)

Proof:

- The diameter provides the lower bound.
- Note \( SE(k) \) is not node-symmetric.
- We have to provide an algorithm for any node \( v \).
- Algorithm has to be without conflicts.
- And we do now show it here in detail.
Theorem:

We have: \( \min b(SE(k)) = b(SE(k)) = 2 \cdot k - 1 \)

Proof:

- The diameter provides the lower bound.
- Note \( SE(k) \) is not node-symmetric.
- We have to provide an algorithm for any node \( v \).
- Algorithm has to be without conflicts.
- And we do now show it here in detail.
Theorem:

We have: $\lfloor 3m/2 \rfloor \leq \min b(BF(m)) = b(BF(m)) \leq 2 \cdot m$

- The diameter gives the lower bound.
- Algorithm will be provided in the following.
Theorem: We have: $\lfloor 3m/2 \rfloor \leq \min b(BF(m)) = b(BF(m)) \leq 2 \cdot m$

- The diameter gives the lower bound.
- Algorithm will be provided in the following.
BF (Idea of proof)

- Distribute the information in two ways:
  - Prefer in the first strategy the cycle-edges.
  - Prefer in the second strategy the cross-edges.

- Split the butterfly into two isomorphic parts.

- Choose for each part a different strategy.

- Distribute in the last phase on the cycles.

\[ \left\lfloor \frac{3m}{2} \right\rfloor \leq \min b(BF(m)) = b(BF(m)) \leq 2 \cdot m \]
Distribute the information in two ways:

- Prefer in the first strategy the cycle-edges.
- Prefer in the second strategy the cross-edges.

Split the butterfly into two isomorph parts.
Choose for each part a different strategy.
Distribute in the last phase on the cycles.
BF (Idea of proof)

- Distribute the information in two ways:
  - Prefer in the first strategy the cycle-edges.
  - Prefer in the second strategy the cross-edges.

- Split the butterfly into two isomorph parts.
- Choose for each part a different strategy.
- Distribute in the last phase on the cycles.

\[
\left\lfloor \frac{3m}{2} \right\rfloor \leq \min b(BF(m)) = b(BF(m)) \leq 2 \cdot m
\]
BF (Idea of proof)

- Distribute the information in two ways:
  - Prefer in the first strategy the cycle-edges.
  - Prefer in the second strategy the cross-edges.

- Split the butterfly into two isomorph parts.
- Choose for each part a different strategy.
- Distribute in the last phase on the cycles.

\[
\left\lfloor \frac{3m}{2} \right\rfloor \leq \min b(BF(m)) = b(BF(m)) \leq 2 \cdot m
\]
BF (Idea of proof)

- Distribute the information in two ways:
  - Prefer in the first strategy the cycle-edges.
  - Prefer in the second strategy the cross-edges.
- Split the butterfly into two isomorph parts.
- **Choose for each part a different strategy.**
- Distribute in the last phase on the cycles.

\[
\left\lceil \frac{3m}{2} \right\rceil \leq \min b(BF(m)) = b(BF(m)) \leq 2 \cdot m
\]
BF (Idea of proof)

- Distribute the information in two ways:
  - Prefer in the first strategy the cycle-edges.
  - Prefer in the second strategy the cross-edges.

- Split the butterfly into two isomorph parts.

- Choose for each part a different strategy.

- Distribute in the last phase on the cycles.

\[ \left\lfloor \frac{3m}{2} \right\rfloor \leq \min b(BF(m)) = b(BF(m)) \leq 2 \cdot m \]
**BF (Proof I)**

- **Splitting of $BF(m)$ in $F_0$ and $F_1$:**
  - $F_0$ has nodes: $\{(l, \alpha 0) | 0 \leq l \leq m - 1, \alpha \in \{0, 1\}^{m-1}\}$.
  - $F_1$ has nodes: $\{(l, \alpha 1) | 0 \leq l \leq m - 1, \alpha \in \{0, 1\}^{m-1}\}$.
  - $F_0$ and $F_1$ are isomorph.

- $\#_0(w)$ denotes the number of 0’en in $w$.
- $\#_1(w)$ denotes the number of 1’en in $w$.

\[
\left\lceil \frac{3m}{2} \right\rceil \leq \min b(BF(m)) = b(BF(m)) \leq 2 \cdot m
\]
Splitting of $BF(m)$ in $F_0$ and $F_1$:

- $F_0$ has nodes: $\{(l, \alpha_0) \mid 0 \leq l \leq m - 1, \alpha \in \{0, 1\}^{m-1}\}$.
- $F_1$ has nodes: $\{(l, \alpha_1) \mid 0 \leq l \leq m - 1, \alpha \in \{0, 1\}^{m-1}\}$.
- $F_0$ and $F_1$ are isomorphic.

$\#_0(w)$ denotes the number of 0'en in $w$.
$\#_1(w)$ denotes the number of 1'en in $w$.

$$\left\lfloor \frac{3m}{2} \right\rfloor \leq \min b(BF(m)) = b(BF(m)) \leq 2 \cdot m$$
\( \lfloor \frac{3m}{2} \rfloor \leq \min b(BF(m)) = b(BF(m)) \leq 2 \cdot m \)

- Splitting of \( BF(m) \) in \( F_0 \) and \( F_1 \):
  - \( F_0 \) has nodes: \( \{ (l, \alpha 0) \mid 0 \leq l \leq m - 1, \alpha \in \{0, 1\}^{m-1} \} \).
  - \( F_1 \) has nodes: \( \{ (l, \alpha 1) \mid 0 \leq l \leq m - 1, \alpha \in \{0, 1\}^{m-1} \} \).
  - \( F_0 \) and \( F_1 \) are isomorph.

- \( \#_0(w) \) denotes the number of 0’en in \( w \).
- \( \#_1(w) \) denotes the number of 1’en in \( w \).
Splitting of $BF(m)$ in $F_0$ and $F_1$:

- $F_0$ has nodes: $\{(l, \alpha 0) \mid 0 \leq l \leq m - 1, \alpha \in \{0, 1\}^{m-1}\}$.
- $F_1$ has nodes: $\{(l, \alpha 1) \mid 0 \leq l \leq m - 1, \alpha \in \{0, 1\}^{m-1}\}$.
- $F_0$ and $F_1$ are isomorph.

- $\#_0(w)$ denotes the number of 0’en in $w$.
- $\#_1(w)$ denotes the number of 1’en in $w$. 

$\lceil \frac{3m}{2} \rceil \leq \min b(BF(m)) = b(BF(m)) \leq 2 \cdot m$
BF (Proof I)

Splitting of $BF(m)$ in $F_0$ and $F_1$:

- $F_0$ has nodes: $\{(l, \alpha 0) \mid 0 \leq l \leq m - 1, \alpha \in \{0, 1\}^{m-1}\}$.
- $F_1$ has nodes: $\{(l, \alpha 1) \mid 0 \leq l \leq m - 1, \alpha \in \{0, 1\}^{m-1}\}$.
- $F_0$ and $F_1$ are isomorphic.

- $\#_0(w)$ denotes the number of 0’en in $w$.
- $\#_1(w)$ denotes the number of 1’en in $w$.

$\lfloor 3m/2 \rfloor \leq \text{minb}(BF(m)) = b(BF(m)) \leq 2 \cdot m$
BF (Proof I)

Splitting of $BF(m)$ in $F_0$ and $F_1$:

- $F_0$ has nodes: $\{(l, 0) | 0 \leq l \leq m - 1, \alpha \in \{0, 1\}^{m-1}\}$.
- $F_1$ has nodes: $\{(l, 1) | 0 \leq l \leq m - 1, \alpha \in \{0, 1\}^{m-1}\}$.
- $F_0$ and $F_1$ are isomorph.
- $\#_0(w)$ denotes the number of 0’en in $w$.
- $\#_1(w)$ denotes the number of 1’en in $w$.

\[
\lfloor \frac{3m}{2} \rfloor \leq \min b(BF(m)) = b(BF(m)) \leq 2 \cdot m
\]
Consider $F_0$: from node $v_0 = (0,00\cdots00)$ exists a unique path of length $m-1$ to $w_0 = (m-1,\alpha0)$ for $\alpha \in \{0,1\}^{m-1}$.

Consider $F_1$: from node $v_1 = (m-1,00\cdots01)$ exists a unique path of length $m-1$ to $w_1 = (0,\alpha1)$ for $\alpha \in \{0,1\}^{m-1}$.

First step of the algorithm $v_0$ informs $v_1$.

Then we use in $F_0$ and $F_1$ two different strategies.
Consider $F_0$: from node $v_0 = (0, 00 \cdots 00)$ exists a unique path of length $m - 1$ to $w_0 = (m - 1, \alpha 0)$ for $\alpha \in \{0, 1\}^{m-1}$.

Consider $F_1$: from node $v_1 = (m - 1, 00 \cdots 01)$ exists a unique path of length $m - 1$ to $w_1 = (0, \alpha 1)$ for $\alpha \in \{0, 1\}^{m-1}$.

First step of the algorithm $v_0$ informs $v_1$.

Then we use in $F_0$ and $F_1$ two different strategies.
Consider $F_0$: from node $v_0 = (0, 00 \cdots 00)$ exists a unique path of length $m - 1$ to $w_0 = (m - 1, \alpha 0)$ for $\alpha \in \{0, 1\}^{m-1}$.

Consider $F_1$: from node $v_1 = (m - 1, 00 \cdots 01)$ exists a unique path of length $m - 1$ to $w_1 = (0, \alpha 1)$ for $\alpha \in \{0, 1\}^{m-1}$.

First step of the algorithm $v_0$ informs $v_1$.

Then we use in $F_0$ and $F_1$ two different strategies.
Consider $F_0$: from node $v_0 = (0, 00 \cdots 00)$ exists a unique path of length $m - 1$ to $w_0 = (m - 1, \alpha 0)$ for $\alpha \in \{0, 1\}^{m-1}$.

Consider $F_1$: from node $v_1 = (m - 1, 00 \cdots 01)$ exists a unique path of length $m - 1$ to $w_1 = (0, \alpha 1)$ for $\alpha \in \{0, 1\}^{m-1}$.

First step of the algorithm $v_0$ informs $v_1$.

Then we use in $F_0$ and $F_1$ two different strategies.
Consider $F_0$: from node $v_0 = (0, 00 \cdots 00)$ exists a unique path of length $m - 1$ to $w_0 = (m - 1, \alpha 0)$ for $\alpha \in \{0, 1\}^{m-1}$.

Consider $F_1$: from node $v_1 = (m - 1, 00 \cdots 01)$ exists a unique path of length $m - 1$ to $w_1 = (0, \alpha 1)$ for $\alpha \in \{0, 1\}^{m-1}$.

First step of the algorithm $v_0$ informs $v_1$.

Then we use in $F_0$ and $F_1$ two different strategies.
Consider $F_0$: from node $v_0 = (0, 00 \cdots 00)$ exists a unique path of length $m - 1$ to $w_0 = (m - 1, \alpha 0)$ for $\alpha \in \{0, 1\}^{m-1}$.

Consider $F_1$: from node $v_1 = (m - 1, 00 \cdots 01)$ exists a unique path of length $m - 1$ to $w_1 = (0, \alpha 1)$ for $\alpha \in \{0, 1\}^{m-1}$.

First step of the algorithm $v_0$ informs $v_1$.

Then we use in $F_0$ and $F_1$ two different strategies.
Aim: Inform in \(\lceil 3m/2 \rceil\) steps the nodes \(w_0 = (m - 1, \alpha 0)\) and \(w_1 = (0, \alpha 1)\) for \(\alpha \in \{0, 1\}^{m-1}\).

If a node \(w_0 = (m - 1, \alpha 0)\) gets informed, then it informs in the next step \(w_1 = (0, \alpha 1)\) (if necessary).

If a node \(w_1 = (0, \alpha 1)\) gets informed, then it informs in the next step \(w_0 = (m - 1, \alpha 0)\) (if necessary).
Aim: Inform in $\lfloor 3m/2 \rfloor$ steps the nodes $w_0 = (m - 1, \alpha 0)$ and $w_1 = (0, \alpha 1)$ for $\alpha \in \{0, 1\}^{m-1}$.

If a node $w_0 = (m - 1, \alpha 0)$ gets informed, then it informs in the next step $w_1 = (0, \alpha 1)$ (if necessary).

If a node $w_1 = (0, \alpha 1)$ gets informed, then it informs in the next step $w_0 = (m - 1, \alpha 0)$ (if necessary).
Aim: Inform in $\lceil 3m/2 \rceil$ steps the nodes $w_0 = (m - 1, \alpha 0)$ and $w_1 = (0, \alpha 1)$ for $\alpha \in \{0, 1\}^{m-1}$.

If a node $w_0 = (m - 1, \alpha 0)$ gets informed, then it informs in the next step $w_1 = (0, \alpha 1)$ (if necessary).

If a node $w_1 = (0, \alpha 1)$ gets informed, then it informs in the next step $w_0 = (m - 1, \alpha 0)$ (if necessary).
In $F_0$ a informed node $(l, \alpha 0)$ sends first to $(l + 1, \alpha 0)$ and then to 
$(l + 1, \alpha (l) 0)$. [$\alpha (l) = \alpha_1 \ldots \bar{\alpha}_l \ldots$]

In $F_1$ a informed node $(l, \alpha 1)$ sends first to $(l + 1, \alpha (l) 1)$ and then to 
$(l + 1, \alpha 1)$.

The time to inform from $v_0 = (0, 00 \cdots 00)$ a node $w_0 = (m - 1, \alpha 0)$ is:

$$1 + \#_0(\alpha) + 2\#_1(\alpha) = m + \#_1(\alpha).$$

The time to inform from $v_1 = (m - 1, 00 \cdots 01)$ a node $w_1 = (0, \alpha 1)$ is:

$$1 + 2\#_0(\alpha) + \#_1(\alpha) = m + \#_0(\alpha).$$
In $F_0$ a informed node $(l, \alpha 0)$ sends first to $(l + 1, \alpha 0)$ and then to $(l + 1, \alpha(l)0)$. [$\alpha(l) = \alpha_1 \ldots \bar{\alpha}_l \ldots$]

In $F_1$ a informed node $(l, \alpha 1)$ sends first to $(l + 1, \alpha(l)1)$ and then to $(l + 1, \alpha 1)$.

The time to inform from $v_0 = (0, 00 \cdots 00)$ a node $w_0 = (m - 1, \alpha 0)$ is: $1 + \#_0(\alpha) + 2\#_1(\alpha) = m + \#_1(\alpha)$.

The time to inform from $v_1 = (m - 1, 00 \cdots 01)$ a node $w_1 = (0, \alpha 1)$ is: $1 + 2\#_0(\alpha) + \#_1(\alpha) = m + \#_0(\alpha)$. 

$\lfloor 3m/2 \rfloor \leq \min b(BF(m)) = b(BF(m)) \leq 2 \cdot m$
BF (Proof IV)

\[ \lfloor \frac{3m}{2} \rfloor \leq \min b(BF(m)) = b(BF(m)) \leq 2 \cdot m \]

- In $F_0$ a informed node $(l, \alpha_0)$ sends first to $(l+1, \alpha_0)$ and then to $(l+1, \alpha(l)0)$. \[ \alpha(l) = \alpha_1 \ldots \alpha_l \ldots \]

- In $F_1$ a informed node $(l, \alpha_1)$ sends first to $(l+1, \alpha(l)1)$ and then to $(l+1, \alpha_1)$.

- The time to inform from $v_0 = (0, 00 \ldots 00)$ a node $w_0 = (m - 1, \alpha_0)$ is:
  \[ 1 + \#_0(\alpha) + 2\#_1(\alpha) = m + \#_1(\alpha). \]

- The time to inform from $v_1 = (m - 1, 00 \ldots 01)$ a node $w_1 = (0, \alpha_1)$ is:
  \[ 1 + 2\#_0(\alpha) + \#_1(\alpha) = m + \#_0(\alpha). \]
In $F_0$ a informed node $(l, \alpha 0)$ sends first to $(l + 1, \alpha 0)$ and then to $(l + 1, \alpha(l)0)$. [$\alpha(l) = \alpha_1 \ldots \bar{\alpha}_l \ldots$]

In $F_1$ a informed node $(l, \alpha 1)$ sends first to $(l + 1, \alpha(l)1)$ and then to $(l + 1, \alpha 1)$.

The time to inform from $v_0 = (0, 00 \cdots 00)$ a node $w_0 = (m - 1, \alpha 0)$ is: $1 + \#_0(\alpha) + 2\#_1(\alpha) = m + \#_1(\alpha)$.

The time to inform from $v_1 = (m - 1, 00 \cdots 01)$ a node $w_1 = (0, \alpha 1)$ is: $1 + 2\#_0(\alpha) + \#_1(\alpha) = m + \#_0(\alpha)$. 

$\lfloor 3m/2 \rfloor \leq \text{minb}(BF(m)) = b(BF(m)) \leq 2 \cdot m$
BF (Proof V)

- **Case 1:** \( m \) is odd:
  - **Case 1.1:** \( \#_1(\alpha) < (m - 1)/2 \):
    Node \( w_0 \) will be informed from \( v_0 \) at time
    \( m + \#_1(\alpha) < (3m - 1)/2 = \lfloor 3m/2 \rfloor \).
    After this \( w_0 \) sends to \( w_1 \).
    \( w_1 \) is informed at time \( \lfloor 3m/2 \rfloor \).
  - **Case 1.2:** \( \#_0(\alpha) < (m - 1)/2 \):
    node \( w_1 \) will be informed from \( v_0 \) at time
    \( m + \#_0(\alpha) < (3m - 1)/2 = \lfloor 3m/2 \rfloor \).
    \( w_0 \) will be informed from \( w_1 \) at time \( \lfloor 3m/2 \rfloor \).
  - **Case 1.3:** \( \#_0(\alpha) = \#_1(\alpha) = (m - 1)/2 \):
    \( w_0 \) is informed at time
    \( m + \#_1(\alpha) = (3m - 1)/2 = \lfloor 3m/2 \rfloor \).
    \( w_1 \) is informed at time \( m + \#_0(\alpha) = (3m - 1)/2 = \lfloor 3m/2 \rfloor \).
**Case 1: m is odd:**

- **Case 1.1: \( \#_1(\alpha) < (m - 1)/2: \)**
  Node \( w_0 \) will be informed from \( v_0 \) at time
  \( m + \#_1(\alpha) < (3m - 1)/2 = \lfloor 3m/2 \rfloor. \)
  After this \( w_0 \) sends to \( w_1. \)
  \( w_1 \) is informed at time \( \lfloor 3m/2 \rfloor. \)

- **Case 1.2: \( \#_0(\alpha) < (m - 1)/2: \)**
  Node \( w_1 \) will be informed from \( v_0 \) at time
  \( m + \#_0(\alpha) < (3m - 1)/2 = \lfloor 3m/2 \rfloor. \)
  \( w_0 \) will be informed from \( w_1 \) at time \( \lfloor 3m/2 \rfloor. \)

- **Case 1.3: \( \#_0(\alpha) = \#_1(\alpha) = (m - 1)/2: \)**
  \( w_0 \) is informed at time
  \( m + \#_1(\alpha) = (3m - 1)/2 = \lfloor 3m/2 \rfloor. \)
  \( w_1 \) is informed at time \( m + \#_0(\alpha) = (3m - 1)/2 = \lfloor 3m/2 \rfloor. \)
BF (Proof V)

Case 1: $m$ is odd:

- Case 1.1: $\#_1(\alpha) < (m - 1)/2$:
  Node $w_0$ will be informed from $v_0$ at time $m + \#_1(\alpha) < (3m - 1)/2 = [3m/2]$.
  After this $w_0$ sends to $w_1$.
  $w_1$ is informed at time $[3m/2]$.

- Case 1.2: $\#_0(\alpha) < (m - 1)/2$:
  Node $w_1$ will be informed from $v_0$ at time $m + \#_0(\alpha) < (3m - 1)/2 = [3m/2]$.
  $w_0$ will be informed from $w_1$ at time $[3m/2]$.

- Case 1.3: $\#_0(\alpha) = \#_1(\alpha) = (m - 1)/2$:
  $w_0$ is informed at time $m + \#_1(\alpha) = (3m - 1)/2 = [3m/2]$.
  $w_1$ is informed at time $m + \#_0(\alpha) = (3m - 1)/2 = [3m/2]$. 

$[3m/2] \leq \min(b(BF(m)) = b(BF(m)) \leq 2 \cdot m$
Case 1: $m$ is odd:

- **Case 1.1:** $\#_1(\alpha) < (m−1)/2$:
  Node $w_0$ will be informed from $v_0$ at time $m + \#_1(\alpha) < (3m − 1)/2 = \lfloor 3m/2 \rfloor$.
  After this, $w_0$ sends to $w_1$.
  $w_1$ is informed at time $\lfloor 3m/2 \rfloor$.

- **Case 1.2:** $\#_0(\alpha) < (m−1)/2$:
  Node $w_1$ will be informed from $v_0$ at time $m + \#_0(\alpha) < (3m − 1)/2 = \lfloor 3m/2 \rfloor$.
  $w_0$ will be informed from $w_1$ at time $\lfloor 3m/2 \rfloor$.

- **Case 1.3:** $\#_0(\alpha) = \#_1(\alpha) = (m−1)/2$:
  $w_0$ is informed at time $m + \#_1(\alpha) = (3m − 1)/2 = \lfloor 3m/2 \rfloor$.
  $w_1$ is informed at time $m + \#_0(\alpha) = (3m − 1)/2 = \lfloor 3m/2 \rfloor$. 

$\lfloor 3m/2 \rfloor \leq \minb(BF(m)) = b(BF(m)) \leq 2 \cdot m$
BF (Proof V)

Case 2: $m$ is even:

- **Case 2.1:** $\#_1(\alpha) \leq (m - 2)/2$:
  node $w_0$ will be informed from $v_0$ at time
  $m + \#_1(\alpha) \leq 3m/2 - 1 < \lfloor 3m/2 \rfloor$.
  Thus node $w_1$ will be informed at time $\lfloor 3m/2 \rfloor$.

- **Case 2.2:** $\#_0(\alpha) \leq (m - 2)/2$:
  node $w_1$ will be informed from $v_0$ at time
  $m + \#_0(\alpha) \leq 3m/2 - 1 < \lfloor 3m/2 \rfloor$.
  Thus node $w_0$ will be informed at time $\lfloor 3m/2 \rfloor$.

In the last phase we distribute the information on the cycles.

- Running time is: $\lceil m/2 \rceil$ rounds.
- Total running time: $\lfloor 3m/2 \rfloor + \lceil m/2 \rceil = 2m$
BF (Proof V)

- **Case 2:** $m$ is even:
  - **Case 2.1:** $\#_1(\alpha) \leq (m - 2)/2$:
    node $w_0$ will be informed from $v_0$ at time
    $m + \#_1(\alpha) \leq 3m/2 - 1 < \lfloor 3m/2 \rfloor$.
    Thus node $w_1$ will be informed at time $\lfloor 3m/2 \rfloor$.
  - **Case 2.2:** $\#_0(\alpha) \leq (m - 2)/2$:
    node $w_1$ will be informed from $v_0$ at time
    $m + \#_0(\alpha) \leq 3m/2 - 1 < \lfloor 3m/2 \rfloor$.
    Thus node $w_0$ will be informed at time $\lfloor 3m/2 \rfloor$.

- In the last phase we distribute the information on the cycles.
- Running time is: $\lceil m/2 \rceil$ rounds.
- Total running time: $\lfloor 3m/2 \rfloor + \lceil m/2 \rceil = 2m$
Case 2: \( m \) is even:

- **Case 2.1:** \( \#_1(\alpha) \leq (m - 2)/2 \):
  node \( w_0 \) will be informed from \( v_0 \) at time \( m + \#_1(\alpha) \leq 3m/2 - 1 < \lfloor 3m/2 \rfloor \).
  Thus node \( w_1 \) will be informed at time \( \lfloor 3m/2 \rfloor \).

- **Case 2.2:** \( \#_0(\alpha) \leq (m - 2)/2 \):
  node \( w_1 \) will be informed from \( v_0 \) at time \( m + \#_0(\alpha) \leq 3m/2 - 1 < \lfloor 3m/2 \rfloor \).
  Thus node \( w_0 \) will be informed at time \( \lfloor 3m/2 \rfloor \).

In the last phase we distribute the information on the cycles.

Running time is: \( \lceil m/2 \rceil \) rounds.

Total running time: \( \lfloor 3m/2 \rfloor + \lceil m/2 \rceil = 2m \)
BF (Proof V)

[3m/2] ≤ min_b(BF(m)) = b(BF(m)) ≤ 2 · m

- **Case 2:** $m$ is even:
  - **Case 2.1:** $\#_1(\alpha) ≤ (m - 2)/2$:
    node $w_0$ will be informed from $v_0$ at time $m + \#_1(\alpha) ≤ 3m/2 - 1 < [3m/2]$.
    Thus node $w_1$ will be informed at time $[3m/2]$.
  - **Case 2.2:** $\#_0(\alpha) ≤ (m - 2)/2$:
    node $w_1$ will be informed from $v_0$ at time $m + \#_0(\alpha) ≤ 3m/2 - 1 < [3m/2]$.
    Thus node $w_0$ will be informed at time $[3m/2]$.

- In the last phase we distribute the information on the cycles.
- Running time is: $[m/2]$ rounds.
- Total running time: $[3m/2] + [m/2] = 2m$
BF (Proof V)

\[ \lceil \frac{3m}{2} \rceil \leq \min b(BF(m)) = b(BF(m)) \leq 2 \cdot m \]

- **Case 2:** \( m \) is even:
  - **Case 2.1:** \( \#_1(\alpha) \leq (m - 2)/2 \):
    node \( w_0 \) will be informed from \( v_0 \) at time
    \[ m + \#_1(\alpha) \leq 3m/2 - 1 < \lceil 3m/2 \rceil. \]
    Thus node \( w_1 \) will be informed at time \( \lceil 3m/2 \rceil \).
  - **Case 2.2:** \( \#_0(\alpha) \leq (m - 2)/2 \):
    node \( w_1 \) will be informed from \( v_0 \) at time
    \[ m + \#_0(\alpha) \leq 3m/2 - 1 < \lceil 3m/2 \rceil. \]
    Thus node \( w_0 \) will be informed at time \( \lceil 3m/2 \rceil \).

- In the last phase we distribute the information on the cycles.
- **Running time is:** \( \lceil m/2 \rceil \) rounds.
- Total running time: \( \lceil 3m/2 \rceil + \lceil m/2 \rceil = 2m \)
Case 2: $m$ is even:

- Case 2.1: $\#_1(\alpha) \leq (m - 2)/2$:
  node $w_0$ will be informed from $v_0$ at time $m + \#_1(\alpha) \leq 3m/2 - 1 < \lfloor 3m/2 \rfloor$.
  Thus node $w_1$ will be informed at time $\lfloor 3m/2 \rfloor$.

- Case 2.2: $\#_0(\alpha) \leq (m - 2)/2$:
  node $w_1$ will be informed from $v_0$ at time $m + \#_0(\alpha) \leq 3m/2 - 1 < \lfloor 3m/2 \rfloor$.
  Thus node $w_0$ will be informed at time $\lfloor 3m/2 \rfloor$.

In the last phase we distribute the information on the cycles.

Running time is: $\lceil m/2 \rceil$ rounds.

Total running time: $\lfloor 3m/2 \rfloor + \lceil m/2 \rceil = 2m$
Theorem:
We have: \( d \leq \min b(DB(d)) = b(DB(d)) \leq \lfloor 3/2 \cdot (d + 1) \rfloor. \)

Proof:
- Idea \((y_1, y_2, \ldots, y_d)\) informs \((y_2, \ldots, y_d, y_1)\) and \((y_2, \ldots, y_d, \overline{y_1})\).
- The order is given by the parity.
- Let \(\alpha = \#_1(y_1, y_2, \ldots, y_d) \mod 2.\)
- \((y_1, y_2, \ldots, y_d)\) informs first \((y_2, \ldots, y_d, \alpha)\) and then \((y_2, \ldots, y_d, \overline{\alpha})\).
- \((0011000)\) informs first \((0110000)\) and then \((0110001)\).
Theorem:
We have: \( d \leq \min b(DB(d)) = b(DB(d)) \leq \left\lfloor \frac{3}{2} \cdot (d + 1) \right\rfloor. \)

Proof:

- Idea \((y_1, y_2, \ldots, y_d)\) informs \((y_2, \ldots, y_d, y_1)\) and \((y_2, \ldots, y_d, \overline{y_1})\).
- The order is given by the parity.
- Let \( \alpha = \#_1(y_1, y_2, \ldots, y_d) \) mod 2.
- \((y_1, y_2, \ldots, y_d)\) informs first \((y_2, \ldots, y_d, \alpha)\) and then \((y_2, \ldots, y_d, \overline{\alpha})\).
- \((0011000)\) informs first \((0110000)\) and then \((0110001)\).
Theorem:
We have: \( d \leq \min b(DB(d)) = b(DB(d)) \leq \lceil 3/2 \cdot (d + 1) \rceil. \)

Proof:
- Idea \((y_1, y_2, \ldots, y_d)\) informs \((y_2, \ldots, y_d, y_1)\) and \((y_2, \ldots, y_d, \overline{y_1})\).
- The order is given by the parity.
- Let \(\alpha = \#_1(y_1, y_2, \ldots, y_d) \mod 2.\)
- \((y_1, y_2, \ldots, y_d)\) informs first \((y_2, \ldots, y_d, \alpha)\) and then \((y_2, \ldots, y_d, \overline{\alpha})\).
- \((0011000)\) informs first \((0110000)\) and then \((0110001)\).
Theorem:

We have: \( d \leq \min b(DB(d)) = b(DB(d)) \leq \lceil 3/2 \cdot (d + 1) \rceil. \)

Proof:

- Idea \((y_1, y_2, \ldots, y_d)\) informs \((y_2, \ldots, y_d, y_1)\) and \((y_2, \ldots, y_d, \overline{y_1})\).
- The order is given by the parity.
- Let \(\alpha = \#_1(y_1, y_2, \ldots, y_d) \mod 2\).
- \((y_1, y_2, \ldots, y_d)\) informs first \((y_2, \ldots, y_d, \alpha)\) and then \((y_2, \ldots, y_d, \overline{\alpha})\).
- \((0011000)\) informs first \((0110000)\) and then \((0110001)\).
Theorem:

We have: \( d \leq \min b(DB(d)) = b(DB(d)) \leq \left\lfloor \frac{3}{2} \cdot (d + 1) \right\rfloor. \)

Proof:

- Idea \((y_1, y_2, \ldots, y_d)\) informs \((y_2, \ldots, y_d, y_1)\) and \((y_2, \ldots, y_d, \overline{y_1})\).
- The order is given by the parity.
- Let \(\alpha = \#_1(y_1, y_2, \ldots, y_d) \mod 2.\)
- \((y_1, y_2, \ldots, y_d)\) informs first \((y_2, \ldots, y_d, \alpha)\) and then \((y_2, \ldots, y_d, \overline{\alpha})\).
- \((0011000)\) informs first \((0110000)\) and then \((0110001)\).
Theorem:

We have: \( d \leq \min b(DB(d)) = b(DB(d)) \leq \left\lfloor \frac{3}{2} \cdot (d + 1) \right\rfloor. \)

Proof:

- Idea \( (y_1, y_2, \ldots, y_d) \) informs \( (y_2, \ldots, y_d, y_1) \) and \( (y_2, \ldots, y_d, \overline{y_1}) \).
- The order is given by the parity.
- Let \( \alpha = \#_1(y_1, y_2, \ldots, y_d) \mod 2. \)
- \( (y_1, y_2, \ldots, y_d) \) informs first \( (y_2, \ldots, y_d, \alpha) \) and then \( (y_2, \ldots, y_d, \overline{\alpha}) \).
- \( (0011000) \) informs first \( (0110000) \) and then \( (0110001) \).
For $k \in \{0, 1\}$ consider the path $P_k$
from $(y_1, y_2, \ldots, y_d)$ to $(z_1, z_2, \ldots, z_{d-1}, z_d)$.

\[(y_1, \ldots, y_d), (y_2, \ldots, y_d, k), (y_3, \ldots, y_d, k, z_1), (y_4, \ldots, y_d, k, z_1, z_2), \ldots\]
\[\ldots, (y_d, k, z_1, \ldots, z_{d-2}), (k, z_1, \ldots, z_{d-1}), (z_1, \ldots, z_{d-1}, z_d)\]

Let $v_{0i} = (y_i, \ldots, y_d, 0, z_1, \ldots, z_{i-2})$ the i-th node on $P_0$.
Let $v_{1i} = (y_i, \ldots, y_d, 1, z_1, \ldots, z_{i-2})$ the i-th node on $P_1$.

We have different times (1 or 2) for sending:

- $(y_i, \ldots, y_d, 0, z_1, \ldots, z_{i-2}) \rightarrow (y_{i+1}, \ldots, y_d, 0, z_1, \ldots, z_{i-2}, z_{i-1})$
- $(y_i, \ldots, y_d, 1, z_1, \ldots, z_{i-2}) \rightarrow (y_{i+1}, \ldots, y_d, 1, z_1, \ldots, z_{i-2}, z_{i-1})$

Thus the sum of running times is on $P_0$ and $P_1$: $3(d + 1)$.
Thus the running time for the broadcast is: $\lfloor 3(d + 1)/2 \rfloor$. 
For $k \in \{0, 1\}$ consider the path $P_k$ from $(y_1, y_2, \ldots, y_d)$ to $(z_1, z_2, \ldots, z_{d-1}, z_d)$.

$$(y_1, \ldots, y_d), (y_2, \ldots, y_d, k), (y_3, \ldots, y_d, k, z_1), (y_4, \ldots, y_d, k, z_1, z_2), \ldots, (y_d, k, z_1, \ldots, z_{d-2}), (k, z_1, \ldots, z_{d-1}), (z_1, \ldots, z_{d-1}, z_d)$$

Let $v_{0i} = (y_i, \ldots, y_d, 0, z_1, \ldots, z_{i-2})$ the $i$-th node on $P_0$.

Let $v_{1i} = (y_i, \ldots, y_d, 1, z_1, \ldots, z_{i-2})$ the $i$-th node on $P_1$.

We have different times (1 or 2) for sending:

- $(y_i, \ldots, y_d, 0, z_1, \ldots, z_{i-2}) \rightarrow (y_{i+1}, \ldots, y_d, 0, z_1, \ldots, z_{i-2}, z_{i-1})$
- $(y_i, \ldots, y_d, 1, z_1, \ldots, z_{i-2}) \rightarrow (y_{i+1}, \ldots, y_d, 1, z_1, \ldots, z_{i-2}, z_{i-1})$

Thus the sum of running times is on $P_0$ and $P_1$: $3(d + 1)$.

Thus the running time for the broadcast is: $\lceil 3(d + 1)/2 \rceil$. 
For $k \in \{0, 1\}$ consider the path $P_k$
from $(y_1, y_2, \ldots, y_d)$ to $(z_1, z_2, \ldots, z_{d-1}, z_d)$.

$$(y_1, \ldots, y_d), (y_2, \ldots, y_d, k), (y_3, \ldots, y_d, k, z_1), (y_4, \ldots, y_d, k, z_1, z_2), \ldots$$
$$\ldots, (y_d, k, z_1, \ldots, z_{d-2}), (k, z_1, \ldots, z_{d-1}), (z_1, \ldots, z_{d-1}, z_d))$$

Let $v_{0i} = (y_i, \ldots, y_d, 0, z_1, \ldots, z_{i-2})$ the $i$-th node on $P_0$.

Let $v_{1i} = (y_i, \ldots, y_d, 1, z_1, \ldots, z_{i-2})$ the $i$-th node on $P_1$.

We have different times (1 or 2) for sending:
- $(y_i, \ldots, y_d, 0, z_1, \ldots, z_{i-2}) \rightarrow (y_{i+1}, \ldots, y_d, 0, z_1, \ldots, z_{i-2}, z_{i-1})$
- $(y_i, \ldots, y_d, 1, z_1, \ldots, z_{i-2}) \rightarrow (y_{i+1}, \ldots, y_d, 1, z_1, \ldots, z_{i-2}, z_{i-1})$.

Thus the sum of running times is on $P_0$ and $P_1$: $3(d + 1)$.

Thus the running time for the broadcast is: $\lfloor 3(d + 1)/2 \rfloor$. 
DB (Proof)

- For $k \in \{0, 1\}$ consider the path $P_k$ from $(y_1, y_2, \ldots, y_d)$ to $(z_1, z_2, \ldots, z_{d-1}, z_d)$.

  $$(y_1, \ldots, y_d), (y_2, \ldots, y_d, k), (y_3, \ldots, y_d, k, z_1), (y_4, \ldots, y_d, k, z_1, z_2), \ldots$$

  $$\ldots, (y_d, k, z_1, \ldots, z_{d-2}), (k, z_1, \ldots, z_{d-1}), (z_1, \ldots, z_{d-1}, z_d)$$

- Let $v_{0i} = (y_i, \ldots, y_d, 0, z_1, \ldots, z_{i-2})$ the i-th node on $P_0$.
- Let $v_{1i} = (y_i, \ldots, y_d, 1, z_1, \ldots, z_{i-2})$ the i-th node on $P_1$.

- We have different times (1 or 2) for sending:
  - $(y_i, \ldots, y_d, 0, z_1, \ldots, z_{i-2}) \rightarrow (y_{i+1}, \ldots, y_d, 0, z_1, \ldots, z_{i-2}, z_{i-1})$
  - $(y_i, \ldots, y_d, 1, z_1, \ldots, z_{i-2}) \rightarrow (y_{i+1}, \ldots, y_d, 1, z_1, \ldots, z_{i-2}, z_{i-1})$.

- Thus the sum of running times is on $P_0$ and $P_1$: $3(d + 1)$.
- Thus the running time for the broadcast is: $\lfloor 3(d + 1)/2 \rfloor$. 
For \( k \in \{0, 1\} \) consider the path \( P_k \) from \((y_1, y_2, \ldots, y_d)\) to \((z_1, z_2, \ldots, z_{d-1}, z_d)\).

\[(y_1, \ldots, y_d), (y_2, \ldots, y_d, k), (y_3, \ldots, y_d, k, z_1), (y_4, \ldots, y_d, k, z_1, z_2), \ldots \]
\[\ldots, (y_d, k, z_1, \ldots, z_{d-2}), (k, z_1, \ldots, z_{d-1}), (z_1, \ldots, z_{d-1}, z_d)\)

Let \( v_{0i} = (y_i, \ldots, y_d, 0, z_1, \ldots, z_{i-2}) \) the i-th node on \( P_0 \).

Let \( v_{1i} = (y_i, \ldots, y_d, 1, z_1, \ldots, z_{i-2}) \) the i-th node on \( P_1 \).

We have different times (1 or 2) for sending:

\begin{itemize}
  \item \((y_i, \ldots, y_d, 0, z_1, \ldots, z_{i-2}) \rightarrow (y_{i+1}, \ldots, y_d, 0, z_1, \ldots, z_{i-2}, z_{i-1})\)
  \item \((y_i, \ldots, y_d, 1, z_1, \ldots, z_{i-2}) \rightarrow (y_{i+1}, \ldots, y_d, 1, z_1, \ldots, z_{i-2}, z_{i-1})\).
\end{itemize}

Thus the sum of running times is on \( P_0 \) and \( P_1 \): \(3(d + 1)\).

Thus the running time for the broadcast is: \(\lfloor 3(d + 1)/2 \rfloor\).
For \( k \in \{0, 1\} \) consider the path \( P_k \) from \((y_1, y_2, \ldots, y_d)\) to \((z_1, z_2, \ldots, z_{d-1}, z_d)\).

\[
(y_1, \ldots, y_d), (y_2, \ldots, y_d, k), (y_3, \ldots, y_d, k, z_1), (y_4, \ldots, y_d, k, z_1, z_2), \ldots \]
\[
\ldots, (y_d, k, z_1, \ldots, z_{d-2}), (k, z_1, \ldots, z_{d-1}), (z_1, \ldots, z_{d-1}, z_d))
\]

Let \( v_{0i} = (y_i, \ldots, y_d, 0, z_1, \ldots, z_{i-2}) \) the i-th node on \( P_0 \).

Let \( v_{1i} = (y_i, \ldots, y_d, 1, z_1, \ldots, z_{i-2}) \) the i-th node on \( P_1 \).

We have different times (1 or 2) for sending:

- \( (y_i, \ldots, y_d, 0, z_1, \ldots, z_{i-2}) \rightarrow (y_{i+1}, \ldots, y_d, 0, z_1, \ldots, z_{i-2}, z_{i-1}) \)
- \( (y_i, \ldots, y_d, 1, z_1, \ldots, z_{i-2}) \rightarrow (y_{i+1}, \ldots, y_d, 1, z_1, \ldots, z_{i-2}, z_{i-1}) \).

Thus the sum of running times is on \( P_0 \) and \( P_1 \): 3\((d + 1)\).

Thus the running time for the broadcast is: \( \lceil 3(d + 1)/2 \rceil \).
DB (Proof)

For $k \in \{0, 1\}$ consider the path $P_k$
from $(y_1, y_2, \ldots, y_d)$ to $(z_1, z_2, \ldots, z_{d-1}, z_d)$.

$$(y_1, \ldots, y_d), (y_2, \ldots, y_d, k), (y_3, \ldots, y_d, k, z_1), (y_4, \ldots, y_d, k, z_1, z_2), \ldots \ldots, (y_d, k, z_1, \ldots, z_{d-2}), (k, z_1, \ldots, z_{d-1}), (z_1, \ldots, z_{d-1}, z_d))$$

Let $v_{0i} = (y_i, \ldots, y_d, 0, z_1, \ldots, z_{i-2})$ the i-th node on $P_0$.
Let $v_{1i} = (y_i, \ldots, y_d, 1, z_1, \ldots, z_{i-2})$ the i-th node on $P_1$.

We have different times (1 or 2) for sending:

- $(y_i, \ldots, y_d, 0, z_1, \ldots, z_{i-2}) \rightarrow (y_{i+1}, \ldots, y_d, 0, z_1, \ldots, z_{i-2}, z_{i-1})$
- $(y_i, \ldots, y_d, 1, z_1, \ldots, z_{i-2}) \rightarrow (y_{i+1}, \ldots, y_d, 1, z_1, \ldots, z_{i-2}, z_{i-1})$.

Thus the sum of running times is on $P_0$ and $P_1$: $3(d + 1)$.
Thus the running time for the broadcast is: $\lfloor 3(d + 1)/2 \rfloor$. 
For $k \in \{0, 1\}$ consider the path $P_k$
from $(y_1, y_2, \ldots, y_d)$ to $(z_1, z_2, \ldots, z_{d-1}, z_d)$.

$$(y_1, \ldots, y_d), (y_2, \ldots, y_d, k), (y_3, \ldots, y_d, k, z_1), (y_4, \ldots, y_d, k, z_1, z_2), \ldots \ldots, (y_d, k, z_1, \ldots, z_{d-2}), (k, z_1, \ldots, z_{d-1}), (z_1, \ldots, z_{d-1}, z_d))$$

Let $v_{0i} = (y_i, \ldots, y_d, 0, z_1, \ldots, z_{i-2})$ the i-th node on $P_0$.
Let $v_{1i} = (y_i, \ldots, y_d, 1, z_1, \ldots, z_{i-2})$ the i-th node on $P_1$.

We have different times (1 or 2) for sending:

- $(y_i, \ldots, y_d, 0, z_1, \ldots, z_{i-2}) \rightarrow (y_{i+1}, \ldots, y_d, 0, z_1, \ldots, z_{i-2}, z_{i-1})$
- $(y_i, \ldots, y_d, 1, z_1, \ldots, z_{i-2}) \rightarrow (y_{i+1}, \ldots, y_d, 1, z_1, \ldots, z_{i-2}, z_{i-1})$.

Thus the sum of running times is on $P_0$ and $P_1$: $3(d + 1)$.
Thus the running time for the broadcast is: $\lfloor 3(d + 1)/2 \rfloor$. 
Degree of the Nodes

**Theorem:**

Let $n \geq 5$ and $G = (V, E)$ be a graph with $n$ nodes:

- If $\Delta(G) = 3$ holds, we have: $b(G) \geq \min b(G) \geq 1.4404 \log(n) - 3$.
- If $\Delta(G) = 4$ holds, we have: $b(G) \geq \min b(G) \geq 1.1374 \log(n) - 2$.

**Proof:**

- Let $A$ be a broadcast-algorithm.
- Let $\text{Broad}^A_i(v_0)$ be the set of nodes, which are informed from $v_0$ by $A$ in $i$ rounds.
- Let $\text{Rec}^A_i(v_0) = \text{Broad}^A_i(v_0) \setminus \text{Broad}^A_{i-1}(v_0)$.
- Let $\text{Rec}^A_0(v_0) = \{v_0\}$.
- We have: $|\text{Broad}^A_i(v_0)| = \sum_{s=0}^{i} |\text{Rec}^A_s(v_0)|$. 
Degree of the Nodes

**Theorem:**

Let $n \geq 5$ and $G = (V, E)$ be a graph with $n$ nodes:

- If $\Delta(G) = 3$ holds, we have: $b(G) \geq \min b(G) \geq 1.4404 \log(n) - 3$.
- If $\Delta(G) = 4$ holds, we have: $b(G) \geq \min b(G) \geq 1.1374 \log(n) - 2$.

**Proof:**

- Let $A$ be a broadcast-algorithm.
- Let $\text{Broad}_i^A(v_0)$ be the set of nodes, which are informed from $v_0$ by $A$ in $i$ rounds.
- Let $\text{Rec}_i^A(v_0) = \text{Broad}_i^A(v_0) \setminus \text{Broad}_{i-1}^A(v_0)$.
- Let $\text{Rec}_0^A(v_0) = \{v_0\}$.
- We have: $|\text{Broad}_i^A(v_0)| = \sum_{s=0}^{i} |\text{Rec}_s^A(v_0)|$. 
Degree of the Nodes

Theorem:
Let $n \geq 5$ and $G = (V, E)$ be a graph with $n$ nodes:

- If $\Delta(G) = 3$ holds, we have: $b(G) \geq \min b(G) \geq 1.4404 \log(n) - 3$.
- If $\Delta(G) = 4$ holds, we have: $b(G) \geq \min b(G) \geq 1.1374 \log(n) - 2$.

Proof:
- Let $A$ be a broadcast-algorithm.
- Let $\text{Broad}_i^A(v_0)$ be the set of nodes, which are informed from $v_0$ by $A$ in $i$ rounds.
- Let $\text{Rec}_i^A(v_0) = \text{Broad}_i^A(v_0) \setminus \text{Broad}_{i-1}^A(v_0)$.
- Let $\text{Rec}_0^A(v_0) = \{v_0\}$.
- We have: $|\text{Broad}_i^A(v_0)| = \sum_{s=0}^{i} |\text{Rec}_s^A(v_0)|$. 
Degree of the Nodes

Theorem:

Let $n \geq 5$ and $G = (V, E)$ be a graph with $n$ nodes:

- If $\Delta(G) = 3$ holds, we have: $b(G) \geq \min b(G) \geq 1.4404 \log(n) - 3$.
- If $\Delta(G) = 4$ holds, we have: $b(G) \geq \min b(G) \geq 1.1374 \log(n) - 2$.

Proof:

- Let $A$ be a broadcast-algorithm.
- Let $Broad_i^A(v_0)$ be the set of nodes, which are informed from $v_0$ by $A$ in $i$ rounds.
- Let $Rec_i^A(v_0) = Broad_i^A(v_0) \setminus Broad_{i-1}^A(v_0)$.
- Let $Rec_0^A(v_0) = \{v_0\}$.
- We have: $|Broad_i^A(v_0)| = \sum_{s=0}^{i} |Rec_s^A(v_0)|$. 
Degree of the Nodes

Theorem:
Let $n \geq 5$ and $G = (V, E)$ be a graph with $n$ nodes:

- If $\Delta(G) = 3$ holds, we have: $b(G) \geq \min b(G) \geq 1.4404 \log(n) - 3$.
- If $\Delta(G) = 4$ holds, we have: $b(G) \geq \min b(G) \geq 1.1374 \log(n) - 2$.

Proof:

- Let $A$ be a broadcast-algorithm.
- Let $\text{Broad}_i^A(v_0)$ be the set of nodes, which are informed from $v_0$ by $A$ in $i$ rounds.
- Let $\text{Rec}_i^A(v_0) = \text{Broad}_i^A(v_0) \setminus \text{Broad}_{i-1}^A(v_0)$.
- Let $\text{Rec}_0^A(v_0) = \{v_0\}$.
- We have: $|\text{Broad}_i^A(v_0)| = \sum_{s=0}^{i} |\text{Rec}_s^A(v_0)|$. 
Degree of the Nodes

**Theorem:** Let $n \geq 5$ and $G = (V, E)$ be a graph with $n$ nodes:

- If $\Delta(G) = 3$ holds, we have: $b(G) \geq \min b(G) \geq 1.4404 \log(n) - 3$.
- IF $\Delta(G) = 4$ holds, we have: $b(G) \geq \min b(G) \geq 1.1374 \log(n) - 2$.

**Proof:**

- Let $A$ be a broadcast-algorithm.
- Let $\text{Broad}_i^A(v_0)$ be the set of nodes, which are informed from $v_0$ by $A$ in $i$ rounds.
- Let $\text{Rec}_i^A(v_0) = \text{Broad}_i^A(v_0) \setminus \text{Broad}_{i-1}^A(v_0)$.
- Let $\text{Rec}_0^A(v_0) = \{v_0\}$.
- We have: $|\text{Broad}_i^A(v_0)| = \sum_{s=0}^{i} |\text{Rec}_s^A(v_0)|$. 

Degree of the Nodes

Theorem:
Let $n \geq 5$ and $G = (V, E)$ be a graph with $n$ nodes:
- If $\Delta(G) = 3$ holds, we have: $b(G) \geq \min b(G) \geq 1.4404 \log(n) - 3$.
- If $\Delta(G) = 4$ holds, we have: $b(G) \geq \min b(G) \geq 1.1374 \log(n) - 2$.

Proof:
- Let $A$ be a broadcast-algorithm.
- Let $\text{Broad}_i^A(v_0)$ be the set of nodes, which are informed from $v_0$ by $A$ in $i$ rounds.
- Let $\text{Rec}_i^A(v_0) = \text{Broad}_i^A(v_0) \setminus \text{Broad}_{i-1}^A(v_0)$.
- Let $\text{Rec}_0^A(v_0) = \{v_0\}$.
- We have: $|\text{Broad}_i^A(v_0)| = \sum_{s=0}^{i} |\text{Rec}_s^A(v_0)|$. 
Degree of the Nodes

**Theorem:**

Let $n \geq 5$ and $G = (V, E)$ be a graph with $n$ nodes:
- If $\Delta(G) = 3$ holds, we have: $b(G) \geq \min b(G) \geq 1.4404 \log(n) - 3$.
- If $\Delta(G) = 4$ holds, we have: $b(G) \geq \min b(G) \geq 1.1374 \log(n) - 2$.

**Proof:**

- Let $A$ be a broadcast-algorithm.
- Let $\text{Broad}^A_i(v_0)$ be the set of nodes, which are informed from $v_0$ by $A$ in $i$ rounds.
- Let $\text{Rec}^A_i(v_0) = \text{Broad}^A_i(v_0) \setminus \text{Broad}^A_{i-1}(v_0)$.
- Let $\text{Rec}^A_0(v_0) = \{v_0\}$.
- We have: $|\text{Broad}^A_i(v_0)| = \sum_{s=0}^{i} |\text{Rec}^A_s(v_0)|$. 
Building the Idea

We consider here only the case $\Delta(G) = 3$. The case $\Delta(G) = 4$ is similar.

- The initial node may send at most three times.
- The initial node sends only in rounds 1, 2, 3.
- Any other nodes will be informed at time $t$ via an edge $e$.
- No further node may be informed via $e$.
- Thus any other node may send at most two times.
- If a node $v$ is informed in round $t$ by $w$, then did $w$ receive the information at round $t-1$ or $t-2$.
- Thus the number of newly informed nodes in round $t > 3$, is at most the number of nodes which got informed in rounds $t-1$ and $t-2$. 
Building the Idea

We consider here only the case $\Delta(G) = 3$. The case $\Delta(G) = 4$ is similar.

- The initial node may send at most three times.
- **The initial node sends only in rounds 1, 2, 3.**
- Any other nodes will be informed at time $t$ via an edge $e$.
- No further node may be informed via $e$.
- Thus any other node may send at most two times.
- If a node $v$ is informed in round $t$ by $w$, then did $w$ receive the information at round $t - 1$ or $t - 2$.
- Thus the number of newly informed nodes in round $t > 3$, is at most the number of nodes which got informed in rounds $t - 1$ and $t - 2$. 
Building the Idea

We consider here only the case $\Delta(G) = 3$. The case $\Delta(G) = 4$ is similar.

- The initial node may send at most three times.
- The initial node sends only in rounds 1, 2, 3.
- Any other nodes will be informed at time $t$ via an edge $e$.
- No further node may be informed via $e$.
- Thus any other node may send at most two times.
- If a node $v$ is informed in round $t$ by $w$, then did $w$ receive the information at round $t-1$ or $t-2$.
- Thus the number of newly informed nodes in round $t > 3$, is at most the number of nodes which got informed in rounds $t-1$ and $t-2$. 
Building the Idea

We consider here only the case $\Delta(G) = 3$. The case $\Delta(G) = 4$ is similar.

- The initial node may send at most three times.
- The initial node sends only in rounds 1, 2, 3.
- Any other nodes will be informed at time $t$ via an edge $e$.
- No further node may be informed via $e$.
- Thus any other node may send at most two times.
- If a node $v$ is informed in round $t$ by $w$, then did $w$ receive the information at round $t - 1$ or $t - 2$.
- Thus the number of newly informed nodes in round $t > 3$, is at most the number of nodes which got informed in rounds $t - 1$ and $t - 2$. 
Building the Idea

We consider here only the case $\Delta(G) = 3$. The case $\Delta(G) = 4$ is similar.

- The initial node may send at most three times.
- The initial node sends only in rounds 1, 2, 3.
- Any other nodes will be informed at time $t$ via an edge $e$.
- No further node may be informed via $e$.
- Thus any other node may send at most two times.
- If a node $v$ is informed in round $t$ by $w$, then did $w$ receive the information at round $t - 1$ or $t - 2$.
- Thus the number of newly informed nodes in round $t > 3$, is at most the number of nodes which got informed in rounds $t - 1$ and $t - 2$. 
Building the Idea

We consider here only the case $\Delta(G) = 3$. The case $\Delta(G) = 4$ is similar.

- The initial node may send at most three times.
- The initial node sends only in rounds 1, 2, 3.
- Any other nodes will be informed at time $t$ via an edge $e$.
- No further node may be informed via $e$.
- Thus any other node may send at most two times.
- If a node $v$ is informed in round $t$ by $w$, then did $w$ receive the information at round $t - 1$ or $t - 2$.
- Thus the number of newly informed nodes in round $t > 3$, is at most the number of nodes which got informed in rounds $t - 1$ and $t - 2$. 
Building the Idea

We consider here only the case $\Delta(G) = 3$. The case $\Delta(G) = 4$ is similar.

- The initial node may send at most three times.
- The initial node sends only in rounds 1, 2, 3.
- Any other nodes will be informed at time $t$ via an edge $e$.
- No further node may be informed via $e$.
- Thus any other node may send at most two times.
- If a node $v$ is informed in round $t$ by $w$, then did $w$ receive the information at round $t - 1$ or $t - 2$.
- Thus the number of newly informed nodes in round $t > 3$, is at most the number of nodes which got informed in rounds $t - 1$ and $t - 2$. 
Proof

- Let $A(i) = |\text{Rec}_i^A(v_0)|$.
- $A(0) = 1$
- $A(1) = 1$
- $A(2) = 2$
- $A(3) = 4$
- $A(i) = A(i - 1) + A(i - 2)$ für $i \geq 4$.
- Show by induction: $A(i) \leq 1.61804^i$ for $i \geq 0$. 
Proof

- Let $A(i) = |Rec_i^A(v_0)|$.
- $A(0) = 1$
- $A(1) = 1$
- $A(2) = 2$
- $A(3) = 4$
- $A(i) = A(i - 1) + A(i - 2)$ für $i \geq 4$.
- Show by induction: $A(i) \leq 1.61804^i$ for $i \geq 0$. 
Proof

- Let $A(i) = |Rec^A_i(v_0)|$.
- $A(0) = 1$
- $A(1) = 1$
- $A(2) = 2$
- $A(3) = 4$
- $A(i) = A(i - 1) + A(i - 2)$ für $i \geq 4$.
- Show by induction: $A(i) \leq 1.61804^i$ for $i \geq 0$. 
Proof

- Let $A(i) = |Rec_i^A(v_0)|$.
- $A(0) = 1$
- $A(1) = 1$
- $A(2) = 2$
- $A(3) = 4$
- $A(i) = A(i - 1) + A(i - 2)$ für $i \geq 4$.
- Show by induction: $A(i) \leq 1.61804^i$ for $i \geq 0$. 
Proof

- Let $A(i) = |Rec_i^A(v_0)|$.
- $A(0) = 1$
- $A(1) = 1$
- $A(2) = 2$
- $A(3) = 4$
- $A(i) = A(i - 1) + A(i - 2)$ für $i \geq 4$.
- Show by induction: $A(i) \leq 1.61804^i$ for $i \geq 0$. 
Proof

- Let $A(i) = |\text{Rec}^A_i(v_0)|$.
- $A(0) = 1$
- $A(1) = 1$
- $A(2) = 2$
- $A(3) = 4$
- $A(i) = A(i - 1) + A(i - 2)$ für $i \geq 4$.
- Show by induction: $A(i) \leq 1.61804^i$ for $i \geq 0$. 
Proof

- Let $A(i) = |Rec^A_i(v_0)|$.
- $A(0) = 1$
- $A(1) = 1$
- $A(2) = 2$
- $A(3) = 4$
- $A(i) = A(i - 1) + A(i - 2)$ für $i \geq 4$.
- Show by induction: $A(i) \leq 1.61804^i$ for $i \geq 0$. 
Proof

- $A(0) = 1 \leq 1 = 1.61804^0$
- $A(1) = 1 \leq 1.61804 = 1.61804^1$
- $A(2) = 2 \leq 2.61805 = 1.61804^2$
- $A(3) = 4 \leq 4.23612 = 1.61804^3$

Induction step ($i \geq 4$):
- We have: $A(j) \leq 1.61804^j$ for any $j \leq i - 1$.
- $A(i) = A(i - 1) + A(i - 2) \leq 1.61804^{i-1} + 1.61804^{i-2} \leq 1.61804^i$
- Note for this: $1.61804 + 1 \leq 1.61804^2$.

Thus we have: $n \leq |\text{Broadcast}_t^A(v_0)| = \sum_{i=0}^{t} |\text{Rec}_i^A(v_0)| \leq \sum_{i=0}^{t} A(i) \leq \sum_{i=0}^{t} 1.61804^i = \frac{1.61804^{t+1} - 1}{1.61804 - 1} \leq 3 \cdot 1.61804^t$

- $t \geq 1.4404 \cdot \log_2 n - 3$.

Proof of the second statement may be done in the same way.
Proof

- $A(0) = 1 \leq 1 = 1.61804^0$
- $A(1) = 1 \leq 1.61804 = 1.61804^1$
- $A(2) = 2 \leq 2.61805 = 1.61804^2$
- $A(3) = 4 \leq 4.23612 = 1.61804^3$
- Induction step ($i \geq 4$):
  - We have: $A(j) \leq 1.61804^j$ for any $j \leq i - 1$.
  - $A(i) = A(i - 1) + A(i - 2) \leq 1.61804^{i-1} + 1.61804^{i-2} \leq 1.61804^i$
  - Note for this: $1.61804 + 1 \leq 1.61804^2$.

Thus we have:

- $n \leq |\text{Broad}_t^A(v_0)| = \sum_{i=0}^t |\text{Rec}_i^A(v_0)| \leq \sum_{i=0}^t A(i) \leq \sum_{i=0}^t 1.61804^i = \frac{1.61804^{t+1} - 1}{1.61804 - 1} \leq 3 \cdot 1.61804^t$
- $t \geq 1.4404 \cdot \log_2 n - 3$.
- Proof of the second statement may be done in the same way.
Proof

- $A(0) = 1 \leq 1 = 1.61804^0$
- $A(1) = 1 \leq 1.61804 = 1.61804^1$
- $A(2) = 2 \leq 2.61805 = 1.61804^2$
- $A(3) = 4 \leq 4.23612 = 1.61804^3$

Induction step ($i \geq 4$):

- We have: $A(j) \leq 1.61804^j$ for any $j \leq i - 1$.
- $A(i) = A(i - 1) + A(i - 2) \leq 1.61804^{i-1} + 1.61804^{i-2} \leq 1.61804^i$
- Note for this: $1.61804 + 1 \leq 1.61804^2$.

Thus we have: $n \leq |\text{Broadcast}_t^A(v_0)| = \sum_{i=0}^{t} |\text{Rec}_i^A(v_0)| \leq \sum_{i=0}^{t} A(i) \leq \sum_{i=0}^{t} 1.61804^i = \frac{1.61804^{t+1} - 1}{1.61804 - 1} \leq 3 \cdot 1.61804^t$

- $t \geq 1.4404 \cdot \log_2 n - 3$.

Proof of the second statement may be done in the same way.
Proof

- $A(0) = 1 \leq 1 = 1.61804^0$
- $A(1) = 1 \leq 1.61804 = 1.61804^1$
- $A(2) = 2 \leq 2.61805 = 1.61804^2$
- $A(3) = 4 \leq 4.23612 = 1.61804^3$

- Induction step ($i \geq 4$):
  - We have: $A(j) \leq 1.61804^j$ for any $j \leq i - 1$.
  - $A(i) = A(i - 1) + A(i - 2) \leq 1.61804^{i-1} + 1.61804^{i-2} \leq 1.61804^i$
  - Note for this: $1.61804 + 1 \leq 1.61804^2$.

- Thus we have: $n \leq |\text{Broadcast}^A_t(v_0)| = \sum_{i=0}^{t} |\text{Rec}^A_i(v_0)| \leq \sum_{i=0}^{t} A(i) \leq \sum_{i=0}^{t} 1.61804^i = \frac{1.61804^{t+1} - 1}{1.61804 - 1} \leq 3 \cdot 1.61804^t$

- $t \geq 1.4404 \cdot \log_2 n - 3$.

- Proof of the second statement may be done in the same way.
Proof

- $A(0) = 1 \leq 1 = 1.61804^0$
- $A(1) = 1 \leq 1.61804 = 1.61804^1$
- $A(2) = 2 \leq 2.61805 = 1.61804^2$
- $A(3) = 4 \leq 4.23612 = 1.61804^3$

Induction step ($i \geq 4$):

- We have: $A(j) \leq 1.61804^j$ for any $j \leq i - 1$.
- $A(i) = A(i - 1) + A(i - 2) \leq 1.61804^{i-1} + 1.61804^{i-2} \leq 1.61804^i$
- Note for this: $1.61804 + 1 \leq 1.61804^2$.

Thus we have: $n \leq |\text{Broadcast}_t^A(v_0)| = \sum_{i=0}^{t} |\text{Rec}_i^A(v_0)| \leq \sum_{i=0}^{t} A(i) \leq \sum_{i=0}^{t} 1.61804^i = \frac{1.61804^{t+1} - 1}{1.61804 - 1} \leq 3 \cdot 1.61804^t$

- $t \geq 1.4404 \cdot \log_2 n - 3$.
- Proof of the second statement may be done in the same way.
Proof

- $A(0) = 1 \leq 1 = 1.61804^0$
- $A(1) = 1 \leq 1.61804 = 1.61804^1$
- $A(2) = 2 \leq 2.61805 = 1.61804^2$
- $A(3) = 4 \leq 4.23612 = 1.61804^3$

Induction step ($i \geq 4$):
- We have: $A(j) \leq 1.61804^j$ for any $j \leq i - 1$.
- $A(i) = A(i - 1) + A(i - 2) \leq 1.61804^{i-1} + 1.61804^{i-2} \leq 1.61804^i$
- Note for this: $1.61804 + 1 \leq 1.61804^2$.

Thus we have: $n \leq |\text{Broadcast}^A_{t}(v_0)| = \sum_{i=0}^{t} |\text{Rec}_i^A(v_0)| \leq \sum_{i=0}^{t} A(i) \leq \sum_{i=0}^{t} 1.61804^i = \frac{1.61804^{t+1} - 1}{1.61804 - 1} \leq 3 \cdot 1.61804^t$

- $t \geq 1.4404 \cdot \log_2 n - 3$.
- Proof of the second statement may be done in the same way.
Proof

- $A(0) = 1 \leq 1 = 1.61804^0$
- $A(1) = 1 \leq 1.61804 = 1.61804^1$
- $A(2) = 2 \leq 2.61805 = 1.61804^2$
- $A(3) = 4 \leq 4.23612 = 1.61804^3$

Induction step ($i \geq 4$):

- We have: $A(j) \leq 1.61804^j$ for any $j \leq i - 1$.
- $A(i) = A(i - 1) + A(i - 2) \leq 1.61804^{i-1} + 1.61804^{i-2} \leq 1.61804^i$
- Note for this: $1.61804 + 1 \leq 1.61804^2$.

Thus we have: $n \leq \left| \text{Broad}_t^A(v_0) \right| = \sum_{i=0}^t \left| \text{Rec}_i^A(v_0) \right| \leq \sum_{i=0}^t A(i) \leq \sum_{i=0}^t 1.61804^i = \frac{1.61804^{i+1}-1}{1.61804-1} \leq 3 \cdot 1.61804^t$

- $t \geq 1.4404 \cdot \log_2 n - 3$.
- Proof of the second statement may be done in the same way.
Proof

- $A(0) = 1 \leq 1 = 1.61804^0$
- $A(1) = 1 \leq 1.61804 = 1.61804^1$
- $A(2) = 2 \leq 2.61805 = 1.61804^2$
- $A(3) = 4 \leq 4.23612 = 1.61804^3$

Induction step ($i \geq 4$):
- We have: $A(j) \leq 1.61804^j$ for any $j \leq i - 1$.
- $A(i) = A(i - 1) + A(i - 2) \leq 1.61804^{i-1} + 1.61804^{i-2} \leq 1.61804^i$
- Note for this: $1.61804 + 1 \leq 1.61804^2$.

Thus we have: $n \leq |\text{Broad}_t^A(v_0)| = \sum_{i=0}^{t} |\text{Rec}_i^A(v_0)| \leq \sum_{i=0}^{t} A(i) \leq \sum_{i=0}^{t} 1.61804^i = \frac{1.61804^{t+1} - 1}{1.61804 - 1} \leq 3 \cdot 1.61804^t$
- $t \geq 1.4404 \cdot \log_2 n - 3$.
- Proof of the second statement may be done in the same way.
Proof

- $A(0) = 1 \leq 1 = 1.61804^0$
- $A(1) = 1 \leq 1.61804 = 1.61804^1$
- $A(2) = 2 \leq 2.61805 = 1.61804^2$
- $A(3) = 4 \leq 4.23612 = 1.61804^3$

Induction step ($i \geq 4$):

- We have: $A(j) \leq 1.61804^j$ for any $j \leq i - 1$.
- $A(i) = A(i - 1) + A(i - 2) \leq 1.61804^{i-1} + 1.61804^{i-2} \leq 1.61804^i$
- Note for this: $1.61804 + 1 \leq 1.61804^2$.

Thus we have: $n \leq |\text{Broadcast}^A_t(v_0)| = \sum_{i=0}^{t} |\text{Rec}^A_i(v_0)| \leq \sum_{i=0}^{t} A(i) \leq \sum_{i=0}^{t} 1.61804^i = \frac{1.61804^{t+1} - 1}{1.61804 - 1} \leq 3 \cdot 1.61804^t$

- $t \geq 1.4404 \cdot \log_2 n - 3$.

Proof of the second statement may be done in the same way.
Proof

- \( A(0) = 1 \leq 1 = 1.61804^0 \)
- \( A(1) = 1 \leq 1.61804 = 1.61804^1 \)
- \( A(2) = 2 \leq 2.61805 = 1.61804^2 \)
- \( A(3) = 4 \leq 4.23612 = 1.61804^3 \)

Induction step \((i \geq 4)\):
- We have: \( A(j) \leq 1.61804^j \) for any \( j \leq i - 1 \).
- \( A(i) = A(i - 1) + A(i - 2) \leq 1.61804^{i-1} + 1.61804^{i-2} \leq 1.61804^i \)
- Note for this: \( 1.61804 + 1 \leq 1.61804^2 \).

Thus we have: \( n \leq |\text{Broadcast}^A_t(v_0)| = \sum_{i=0}^t |\text{Rec}_i^A(v_0)| \leq \sum_{i=0}^t A(i) \leq \sum_{i=0}^t 1.61804^i = \frac{1.61804^{t+1} - 1}{1.61804 - 1} \leq 3 \cdot 1.61804^t \)

- \( t \geq 1.4404 \cdot \log_2 n - 3. \)

Proof of the second statement may be done in the same way.
Proof

- $A(0) = 1 \leq 1 = 1.61804^0$
- $A(1) = 1 \leq 1.61804 = 1.61804^1$
- $A(2) = 2 \leq 2.61805 = 1.61804^2$
- $A(3) = 4 \leq 4.23612 = 1.61804^3$
- Induction step ($i \geq 4$):
  - We have: $A(j) \leq 1.61804^j$ for any $j \leq i - 1$.
  - $A(i) = A(i - 1) + A(i - 2) \leq 1.61804^{i-1} + 1.61804^{i-2} \leq 1.61804^i$
  - Note for this: $1.61804 + 1 \leq 1.61804^2$.
- Thus we have: $n \leq |\text{Broad}_t^A(v_0)| = \sum_{i=0}^t |\text{Rec}_i^A(v_0)| \leq \sum_{i=0}^t A(i) \leq \sum_{i=0}^t 1.61804^i = \frac{1.61804^{t+1} - 1}{1.61804 - 1} \leq 3 \cdot 1.61804^t$
- $t \geq 1.4404 \cdot \log_2 n - 3$.
- Proof of the second statement may be done in the same way.
More Results

**Consequence:**

\[ b(DB_k) \geq \min b(DB_k) \geq 1.1374 \cdot k - 2 \]

**Theorem:**

\[ b(BF_m) = \min b(BF_m) > 1.7396m \text{ for large enough } m. \]

Idea of Proof: Check the number of nodes in distance \( k \).

**Theorem:**

\[ b(DB_m) > 1.3042m \text{ for large enough } m. \]

Idea of Proof: Check the number of nodes in distance \( k \).
More Results

**Consequence:**

\[ b(DB_k) \geq \min b(DB_k) \geq 1.1374 \cdot k - 2 \]

**Theorem:**

\[ b(BF_m) = \min b(BF_m) > 1.7396m \text{ for large enough } m. \]

Idea of Proof: Check the number of nodes in distance \( k \).

**Theorem:**

\[ b(DB_m) > 1.3042m \text{ for large enough } m. \]

Idea of Proof: Check the number of nodes in distance \( k \).
More Results

Consequence:
\[ b(DB_k) \geq \min b(DB_k) \geq 1.1374 \cdot k - 2 \]

Theorem:
\[ b(BF_m) = \min b(BF_m) > 1.7396m \text{ for large enough } m. \]

Idea of Proof: Check the number of nodes in distance \( k \).

Theorem:
\[ b(DB_m) > 1.3042m \text{ for large enough } m. \]

Idea of Proof: Check the number of nodes in distance \( k \).
## Overview

| Graph | $|V|$ | Diameter | Lower Bound | Upper Bound |
|-------|-----|----------|-------------|-------------|
| $K_n$ | $n$ | 1        | $\lceil \log_2 n \rceil$ | $\lceil \log_2 n \rceil$ |
| $HQ_k$ | $2^k$ | $k$ | $k$ | $k$ |
| $CCC_k$ | $k \cdot 2^k$ | $\lceil 5k/2 \rceil - 2$ | $\lceil 5k/2 \rceil - 2$ | $\lceil 5k/2 \rceil - 2$ |
| $SE_k$ | $2^k$ | $2k - 1$ | $2k - 1$ | $2k - 1$ |
| $DB_k$ | $2^k$ | $k$ | $1.4404k$ | $\frac{3}{2}(k + 1)$ |
| $BF_k$ | $k \cdot 2^k$ | $\lceil 3k/2 \rceil$ | $1.7609k$ | $2k - \frac{1}{2} \log \log k + c$ |
Definition

In edge-disjoint-path communication the information is passed on a set of edge-disjoint paths between the endpoint of each path. A sending or receiving node may not forward any information at the same time.

A edge-disjoint communication algorithm for $G$ is a sequence of rounds $A_1, A_2, \ldots, A_k$, where each $A_i$ is a correct edge-disjoint-path communication.

- $edp-b(G) =$ maximal time to broadcast in edp-mode in $G$.
- $edp-a(G) =$ maximal time to accumulate in edp-mode in $G$.
- $edp-r_1(G) =$ minimal time to gossip in 1-way edp-mode in $G$.
- $edp-r_2(G) =$ minimal time to gossip in 2-way edp-mode in $G$. 
Definition

In edge-disjoint-path communication the information is passed on a set of edge-disjoint paths between the endpoint of each path. A sending or receiving node may not forward any information at the same time. A edge-disjoint communication algorithm for $G$ is a sequence of rounds $A_1, A_2, \ldots, A_k$, where each $A_i$ is a correct edge-disjoint-path communication.

- $edp-b(G) =$ maximal time to broadcast in edp-mode in $G$.
- $edp-a(G) =$ maximal time to accumulate in edp-mode in $G$.
- $edp-r_1(G) =$ minimal time to gossip in 1-way edp-mode in $G$.
- $edp-r_2(G) =$ minimal time to gossip in 2-way edp-mode in $G$. 
Definition

In edge-disjoint-path communication the information is passed on a set of edge-disjoint paths between the endpoint of each path. A sending or receiving node may not forward any information at the same time.

A edge-disjoint communication algorithm for $G$ is a sequence of rounds $A_1, A_2, \ldots, A_k$, where each $A_i$ is a correct edge-disjoint-path communication.

- $edp-b(G) =$ maximal time to broadcast in edp-mode in $G$.
- $edp-a(G) =$ maximal time to accumulate in edp-mode in $G$.
- $edp-r_1(G) =$ minimal time to gossip in 1-way edp-mode in $G$.
- $edp-r_2(G) =$ minimal time to gossip in 2-way edp-mode in $G$. 
Definition

In edge-disjoint-path communication the information is passed on a set of edge-disjoint paths between the endpoint of each path. A sending or receiving node may not forward any information at the same time. A edge-disjoint communication algorithm for $G$ is a sequence of rounds $A_1, A_2, \ldots, A_k$, where each $A_i$ is a correct edge-disjoint-path communication.

- $edp\text{-}b(G) =$ maximal time to broadcast in edp-mode in $G$.
- $edp\text{-}a(G) =$ maximal time to accumulate in edp-mode in $G$.
- $edp\text{-}r_1(G) =$ minimal time to gossip in 1-way edp-mode in $G$.
- $edp\text{-}r_2(G) =$ minimal time to gossip in 2-way edp-mode in $G$. 
Idea \( (edp-a(G)) \)

**Definition**

A set of vertices \( K \subset V \) is called **knowledge set**, if the pieces of information residing in the vertices of \( K \) form the cumulative message.

**Definition**

Let \( T = (V, E) \) be some tree, we will refer to any vertex of degree \( > 2 \) in \( T \) as a critical vertex, while all other vertices are called non-critical.

- Collect within two rounds all pieces of information in a subset \( K \) of non-critical vertices with \( |S| \leq \lfloor n/2 \rfloor \).

- In each of \( \lceil \log_2 n \rceil - 1 \) communication rounds, reduce the size of a given knowledge set \( K \) by a factor of two.
Idea ($edp\cdot a(G)$)

**Definition**

A set of vertices $K \subset V$ is called a **knowledge set**, if the pieces of information residing in the vertices of $K$ form the cumulative message.

**Definition**

Let $T = (V, E)$ be some tree, we will refer to any vertex of degree $> 2$ in $T$ as a **critical vertex**, while all other vertices are called **non-critical**.

- Collect within two rounds all pieces of information in a subset $K$ of non-critical vertices with $|S| \leq \lfloor n/2 \rfloor$.
- In each of $\lceil \log_2 n \rceil - 1$ communication rounds, reduce the size of a given knowledge set $K$ by a factor of two.
Idea \((edp-a(G))\)

**Definition**

A set of vertices \(K \subseteq V\) is called \textbf{knowledge set}, if the pieces of information residing in the vertices of \(K\) form the cumulative message.

**Definition**

Let \(T = (V, E)\) be some tree, we will refer to any vertex of degree \(> 2\) in \(T\) as a critical vertex, while all other vertices are called non-critical.

- Collect within two rounds all pieces of information in a subset \(K\) of non-critical vertices with \(|S| \leq \lfloor n/2 \rfloor\).
- In each of \(\lceil \log_2 n \rceil - 1\) communication rounds, reduce the size of a given knowledge set \(K\) by a factor of two.
Idea $\text{edp-a}(G)$

**Definition**

A set of vertices $K \subset V$ is called knowledge set, if the pieces of information residing in the vertices of $K$ form the cumulative message.

**Definition**

Let $T = (V, E)$ be some tree, we will refer to any vertex of degree $> 2$ in $T$ as a critical vertex, while all other vertices are called non-critical.

- Collect within two rounds all pieces of information in a subset $K$ of non-critical vertices with $|S| \leq \lfloor n/2 \rfloor$.
- In each of $[\log_2 n] - 1$ communication rounds, reduce the size of a given knowledge set $K$ by a factor of two.
Lower Bound

Lemma:

Let $G = (V, E)$ a graph with $n$ nodes. Then we have:

$$r(G) \geq r_2(G) \geq \begin{cases} 
\lceil \log_2 n \rceil & n \text{ even,} \\
\lceil \log_2 n \rceil + 1 & n \text{ odd.}
\end{cases}$$

Proof: Only the case, where $n$ is odd, has to be proven.

- Show: $r_2(G) \geq \lceil \log_2 n \rceil + 1$.

- Let $A$ be a communication-algorithm for the gossip-problem. $A$ has communication rounds (matchings) $E_1, E_2, \ldots, E_k$.

- Show by induction: After $i$ rounds has each node at most $2^i$ pieces of information.
  - $i = 0$: Each node has $2^0 = 1$ pieces of information.
  - $i - 1 \rightarrow i$: at most $2^{i-1} + 2^{i-1} = 2^i$ pieces of information may be collected by any node.

- In round $k$ is at least one node $v$ inactive.

- $v$ has after $k$ rounds at most $2^{k-1}$ pieces of information.
Lower Bound

Lemma:

Let $G = (V, E)$ a graph with $n$ nodes. Then we have:

\[ r(G) \geq r_2(G) \geq \begin{cases} 
\lceil \log_2 n \rceil & n \text{ even,} \\
\lceil \log_2 n \rceil + 1 & n \text{ odd.}
\end{cases} \]

Proof: Only the case, where $n$ is odd, has to be proven.

- **Show**: $r_2(G) \geq \lceil \log_2 n \rceil + 1$.

- Let $A$ be a communication-algorithm for the gossip-problem. $A$ has communication rounds (matchings) $E_1, E_2, \ldots, E_k$.

- **Show by induction**: After $i$ rounds has each node at most $2^i$ pieces of information.
  - $i = 0$: Each node has $2^0 = 1$ pieces of information.
  - $i - 1 \rightarrow i$: at most $2^{i-1} + 2^{i-1} = 2^i$ pieces of information may be collected by any node.

- In round $k$ is at least one node $v$ inactive.

- $v$ has after $k$ rounds at most $2^{k-1}$ pieces of information.
Lower Bound

Lemma:
Let \( G = (V, E) \) a graph with \( n \) nodes. Then we have:

\[
r(G) \geq r_2(G) \geq \begin{cases} 
\lceil \log_2 n \rceil & n \text{ even,} \\
\lceil \log_2 n \rceil + 1 & n \text{ odd.}
\end{cases}
\]

Proof: Only the case, where \( n \) is odd, has to be proven.

- Show: \( r_2(G) \geq \lceil \log_2 n \rceil + 1 \).

- Let \( A \) be a communication-algorithm for the gossip-problem. \( A \) has communication rounds (matchings) \( E_1, E_2, \cdots, E_k \).

- Show by induction: After \( i \) rounds has each node at most \( 2^i \) pieces of information.
  - \( i = 0 \): Each node has \( 2^0 = 1 \) pieces of information.
  - \( i - 1 \rightarrow i \): at most \( 2^{i-1} + 2^{i-1} = 2^i \) pieces of information may be collected by any node.

- In round \( k \) is at least one node \( v \) inactive.

- \( v \) has after \( k \) rounds at most \( 2^{k-1} \) pieces of information.
Lemma:

Let $G = (V, E)$ a graph with $n$ nodes. Then we have:

$$r(G) \geq r_2(G) \geq \begin{cases} \lceil \log_2 n \rceil & n \text{ even}, \\ \lceil \log_2 n \rceil + 1 & n \text{ odd}. \end{cases}$$

Proof: Only the case, where $n$ is odd, has to be proven.

- **Show:** $r_2(G) \geq \lceil \log_2 n \rceil + 1$.

- Let $A$ be a communication-algorithm for the gossip-problem. $A$ has communication rounds (matchings) $E_1, E_2, \cdots, E_k$.

- **Show by induction:** After $i$ rounds has each node at most $2^i$ pieces of information.
  
  - $i = 0$: Each node has $2^0 = 1$ pieces of information.
  - $i - 1 \rightarrow i$: at most $2^{i-1} + 2^{i-1} = 2^i$ pieces of information may be collected by any node.

- In round $k$ is at least one node $v$ inactive.

- $v$ has after $k$ rounds at most $2^{k-1}$ pieces of information.
Lower Bound

Lemma:
Let $G = (V, E)$ a graph with $n$ nodes. Then we have:

$$r(G) \geq r_2(G) \geq \begin{cases} \lceil \log_2 n \rceil & n \text{ even}, \\ \lceil \log_2 n \rceil + 1 & n \text{ odd}. \end{cases}$$

Proof: Only the case, where $n$ is odd, has to be proven.

- Show: $r_2(G) \geq \lceil \log_2 n \rceil + 1$.
- Let $A$ be a communication-algorithm for the gossip-problem. $A$ has communication rounds (matchings) $E_1, E_2, \ldots, E_k$.
- Show by induction: After $i$ rounds has each node at most $2^i$ pieces of information.
  - $i = 0$: Each node has $2^0 = 1$ pieces of information.
  - $i - 1 \rightarrow i$: at most $2^{i-1} + 2^{i-1} = 2^i$ pieces of information may be collected by any node.
- In round $k$ is at least one node $v$ inactive.
- $v$ has after $k$ rounds at most $2^{k-1}$ pieces of information.
Lower Bound

Lemma:
Let $G = (V, E)$ a graph with $n$ nodes. Then we have:

$$r(G) \geq r_2(G) \geq \begin{cases} \lceil \log_2 n \rceil & n \text{ even}, \\ \lceil \log_2 n \rceil + 1 & n \text{ odd}. \end{cases}$$

Proof: Only the case, where $n$ is odd, has to be proven.

- Show: $r_2(G) \geq \lceil \log_2 n \rceil + 1$.

- Let $A$ be a communication-algorithm for the gossip-problem. $A$ has communication rounds (matchings) $E_1, E_2, \cdots, E_k$.

- Show by induction: After $i$ rounds has each node at most $2^i$ pieces of information.
  - $i = 0$: Each node has $2^0 = 1$ pieces of information.
  - $i - 1 \rightarrow i$: at most $2^{i-1} + 2^{i-1} = 2^i$ pieces of information may be collected by any node.

- In round $k$ is at least one node $v$ inactive.

- $v$ has after $k$ rounds at most $2^{k-1}$ pieces of information.
Lower Bound

Lemma:

Let $G = (V, E)$ a graph with $n$ nodes. Then we have:

$$r(G) \geq r_2(G) \geq \begin{cases} \lceil \log_2 n \rceil & n \text{ even,} \\ \lceil \log_2 n \rceil + 1 & n \text{ odd.} \end{cases}$$

Proof: Only the case, where $n$ is odd, has to be proven.

- Show: $r_2(G) \geq \lceil \log_2 n \rceil + 1$.
- Let $A$ be a communication-algorithm for the gossip-problem. $A$ has communication rounds (matchings) $E_1, E_2, \ldots, E_k$.
- Show by induction: After $i$ rounds has each node at most $2^i$ pieces of information.
  - $i = 0$: Each node has $2^0 = 1$ pieces of information.
  - $i - 1 \rightarrow i$: at most $2^{i-1} + 2^{i-1} = 2^i$ pieces of information may be collected by any node.
- In round $k$ is at least one node $v$ inactive.
- $v$ has after $k$ rounds at most $2^{k-1}$ pieces of information.
Lemma:

Let $G = (V, E)$ a graph with $n$ nodes. Then we have:

$$r(G) \geq r_2(G) \geq \begin{cases} 
\lceil \log_2 n \rceil & \text{n even}, \\
\lceil \log_2 n \rceil + 1 & \text{n odd}.
\end{cases}$$

Proof: Only the case, where $n$ is odd, has to be proven.

- **Show:** $r_2(G) \geq \lceil \log_2 n \rceil + 1$.

- Let $A$ be a communication-algorithm for the gossip-problem.
  $A$ has communication rounds (matchings) $E_1, E_2, \cdots, E_k$.

- **Show by induction:** After $i$ rounds has each node at most $2^i$ pieces of information.
  
  - $i = 0$: Each node has $2^0 = 1$ pieces of information.
  
  - $i - 1 \rightarrow i$: at most $2^{i-1} + 2^{i-1} = 2^i$ pieces of information may be collected by any node.

- In round $k$ is at least one node $v$ inactive.

- $v$ has after $k$ rounds at most $2^{k-1}$ pieces of information.
**Lemma:**

For any graph $G = (V, E)$ with $|V| = n$ we have:

- $r(G) \leq 2n - 2$, and
- $r_2(G) \leq 2n - 3$.

**Proof:** Follows from the following known statements:

- $\minb(G) \leq n - 1$ for any graph $G = (V, E)$ with $|V| = n$.
- $r(G) \leq 2 \cdot \minb(G)$
- $r_2(G) \leq 2 \cdot \minb(G) - 1$
Lemma:
For any graph $G = (V, E)$ with $|V| = n$ we have:
- $r(G) \leq 2n - 2$, and
- $r_2(G) \leq 2n - 3$.

Proof: Follows from the following known statements:
- $\minb(G) \leq n - 1$ for any graph $G = (V, E)$ with $|V| = n$.
- $r(G) \leq 2 \cdot \minb(G)$
- $r_2(G) \leq 2 \cdot \minb(G) - 1$
Lemma:

For any graph $G = (V, E)$ with $|V| = n$ we have:

- $r(G) \leq 2n - 2$, and
- $r_2(G) \leq 2n - 3$.

Proof: Follows from the following known statements:

- $\minb(G) \leq n - 1$ for any graph $G = (V, E)$ with $|V| = n$.
- $r(G) \leq 2 \cdot \minb(G)$
- $r_2(G) \leq 2 \cdot \minb(G) - 1$
Lemma:

For any graph \( G = (V, E) \) with \( |V| = n \) we have:
- \( r(G) \leq 2n - 2 \), and
- \( r_2(G) \leq 2n - 3 \).

Proof: Follows from the following known statements:
- \( \minb(G) \leq n - 1 \) for any graph \( G = (V, E) \) with \( |V| = n \).
- \( r(G) \leq 2 \cdot \minb(G) \)
- \( r_2(G) \leq 2 \cdot \minb(G) - 1 \)
Simple Algorithm

Lemma:
For any graph $G = (V, E)$ with $|V| = n$ we have:
- $r(G) \leq 2n - 2$, and
- $r_2(G) \leq 2n - 3$.

Proof: Follows from the following known statements:
- $\minb(G) \leq n - 1$ for any graph $G = (V, E)$ with $|V| = n$.
- $r(G) \leq 2 \cdot \minb(G)$
- $r_2(G) \leq 2 \cdot \minb(G) - 1$
Lemma:
For any graph $G = (V, E)$ with $|V| = n$ we have:
- $r(G) \leq 2n - 2$, and
- $r_2(G) \leq 2n - 3$.

Proof: Follows from the following known statements:
- $\minb(G) \leq n - 1$ for any graph $G = (V, E)$ with $|V| = n$.
- $r(G) \leq 2 \cdot \minb(G)$
- $r_2(G) \leq 2 \cdot \minb(G) - 1$
Lemma:

For any graph $G = (V, E)$ with $|V| = n$ we have:
- $r(G) \leq 2n - 2$, and
- $r_2(G) \leq 2n - 3$.

Proof: Follows from the following known statements:
- $\minb(G) \leq n - 1$ for any graph $G = (V, E)$ with $|V| = n$.
- $r(G) \leq 2 \cdot \minb(G)$
- $r_2(G) \leq 2 \cdot \minb(G) - 1$
Simple Algorithm (Continuation)

Lemma:
We have:
- \( r(T_k(1)) = 2k \)
- \( r_2(T_k(1)) = 2k - 1 \)

Proof:
- Show: \( r(T_k(1)) \geq 2k \).
- \( r(T_k(1)) \) has one root and \( k \) leaves.
- The maximal matching is 1.
- In each round is only one leaf active.
- Each leaf has to send at least once.
- Each leaf has to receive at least once.
- Thus in total \( 2k \) rounds necessary.
- \( r_2(T_k(1)) \geq 2k - 1 \), is a simple exercise.
Simple Algorithm (Continuation)

Lemma:

We have:
1. \( r(T_k(1)) = 2k \)
2. \( r_2(T_k(1)) = 2k - 1 \)

Proof:

1. Show: \( r(T_k(1)) \geq 2k \).
2. \( r(T_k(1)) \) has one root and \( k \) leaves.
3. The maximal matching is 1.
4. In each round is only one leaf active.
5. Each leaf has to send at least once.
6. Each leaf has to receive at least once.
7. Thus in total \( 2k \) rounds necessary.
8. \( r_2(T_k(1)) \geq 2k - 1 \), is a simple exercise.
Simple Algorithm (Continuation)

Lemma:

We have:
- \( r(T_k(1)) = 2k \)
- \( r_2(T_k(1)) = 2k - 1 \)

Proof:
- Show: \( r(T_k(1)) \geq 2k \).
- \( r(T_k(1)) \) has one root and \( k \) leaves.
- The maximal matching is 1.
- In each round is only one leaf active.
- Each leaf has to send at least once.
- Each leaf has to receive at least once.
- Thus in total \( 2k \) rounds necessary.
- \( r_2(T_k(1)) = 2k - 1 \), is a simple exercise.
Simple Algorithm (Continuation)

Lemma:
We have:
- $r(T_k(1)) = 2k$
- $r_2(T_k(1)) = 2k - 1$

Proof:
- **Show**: $r(T_k(1)) \geq 2k$.
- $r(T_k(1))$ has one root and $k$ leaves.
- The maximal matching is 1.
- In each round is only one leaf active.
- Each leaf has to send at least once.
- Each leaf has to receive at least once.
- Thus in total $2k$ rounds necessary.
- $r_2(T_k(1)) \geq 2k - 1$, is a simple exercise.
Simple Algorithm (Continuation)

Lemma:

We have:
- \( r(T_k(1)) = 2k \)
- \( r_2(T_k(1)) = 2k - 1 \)

Proof:
- Show: \( r(T_k(1)) \geq 2k \).
- \( r(T_k(1)) \) has one root and \( k \) leaves.
- The maximal matching is 1.
- In each round is only one leaf active.
- Each leaf has to send at least once.
- Each leaf has to receive at least once.
- Thus in total \( 2k \) rounds necessary.
- \( r_2(T_k(1)) \geq 2k - 1 \), is a simple exercise.
Simple Algorithm (Continuation)

Lemma:

We have:

- \( r(T_k(1)) = 2k \)
- \( r_2(T_k(1)) = 2k - 1 \)

Proof:

- Show: \( r(T_k(1)) \geq 2k \).
- \( r(T_k(1)) \) has one root and \( k \) leaves.
- The maximal matching is 1.
- In each round is only one leaf active.
- Each leaf has to send at least once.
- Each leaf has to receive at least once.
- Thus in total \( 2k \) rounds necessary.
- \( r_2(T_k(1)) \geq 2k - 1 \), is a simple exercise.
Simple Algorithm (Continuation)

Lemma:

We have:

- $r(T_k(1)) = 2k$
- $r_2(T_k(1)) = 2k - 1$

Proof:

- **Show:** $r(T_k(1)) \geq 2k$.
- $r(T_k(1))$ has one root and $k$ leaves.
- The maximal matching is 1.
- **In each round is only one leaf active.**
- Each leaf has to send at least once.
- Each leaf has to receive at least once.
- Thus in total $2k$ rounds necessary.
- $r_2(T_k(1)) \geq 2k - 1$, is a simple exercise.
Simple Algorithm (Continuation)

Lemma:

We have:
- \( r(T_k(1)) = 2k \)
- \( r_2(T_k(1)) = 2k - 1 \)

Proof:

- Show: \( r(T_k(1)) \geq 2k. \)
- \( r(T_k(1)) \) has one root and \( k \) leaves.
- The maximal matching is 1.
- In each round is only one leaf active.
- Each leaf has to send at least once.
- Each leaf has to receive at least once.
- Thus in total \( 2k \) rounds necessary.
- \( r_2(T_k(1)) \geq 2k - 1, \) is a simple exercise.
Simple Algorithm (Continuation)

Lemma:
We have:
- \( r(T_k(1)) = 2k \)
- \( r_2(T_k(1)) = 2k - 1 \)

Proof:
- Show: \( r(T_k(1)) \geq 2k \).
- \( r(T_k(1)) \) has one root and \( k \) leaves.
- The maximal matching is 1.
- In each round is only one leaf active.
- Each leaf has to send at least once.
- Each leaf has to receive at least once.
- Thus in total \( 2k \) rounds necessary.
- \( r_2(T_k(1)) \geq 2k - 1 \), is a simple exercise.
Simple Algorithm (Continuation)

Lemma:

We have:
- \( r(T_k(1)) = 2k \)
- \( r_2(T_k(1)) = 2k - 1 \)

Proof:
- Show: \( r(T_k(1)) \geq 2k \).
- \( r(T_k(1)) \) has one root and \( k \) leaves.
- The maximal matching is 1.
- In each round is only one leaf active.
- Each leaf has to send at least once.
- Each leaf has to receive at least once.
- Thus in total \( 2k \) rounds necessary.
- \( r_2(T_k(1)) \geq 2k - 1 \), is a simple exercise.
Simple Algorithm (Continuation)

Lemma:
We have:
- \( r(T_k(1)) = 2k \)
- \( r_2(T_k(1)) = 2k - 1 \)

Proof:
- Show: \( r(T_k(1)) \geq 2k \).
- \( r(T_k(1)) \) has one root and \( k \) leaves.
- The maximal matching is 1.
- In each round is only one leaf active.
- Each leaf has to send at least once.
- Each leaf has to receive at least once.
- Thus in total \( 2k \) rounds necessary.
- \( r_2(T_k(1)) \geq 2k - 1 \), is a simple exercise.
Gossip on Lines

Theorem:

We have:
- \( r_2(L(n)) = n - 1 \) for any even number \( n \geq 2 \),
- \( r_2(L(n)) = n \) for any odd number \( n \geq 3 \),
- \( r(L(n)) = n \) for any even number \( n \geq 2 \) and
- \( r(L(n)) = n + 1 \) for any odd number \( n \geq 3 \).

Proof:
- All are more or less easy.
Gossip on Lines

Theorem:

We have:
- \( r_2(L(n)) = n - 1 \) for any even number \( n \geq 2 \),
- \( r_2(L(n)) = n \) for any odd number \( n \geq 3 \),
- \( r(L(n)) = n \) for any even number \( n \geq 2 \) and
- \( r(L(n)) = n + 1 \) for any odd number \( n \geq 3 \).

Proof:
- All are more or less easy.
Gossip on Lines

Theorem:

We have:

- \( r_2(L(n)) = n - 1 \) for any even number \( n \geq 2 \),
- \( r_2(L(n)) = n \) for any odd number \( n \geq 3 \),
- \( r(L(n)) = n \) for any even number \( n \geq 2 \) and
- \( r(L(n)) = n + 1 \) for any odd number \( n \geq 3 \).

Proof:

- All are more or less easy.
Gossip on Lines

Theorem:
We have:
- \( r_2(L(n)) = n - 1 \) for any even number \( n \geq 2 \),
- \( r_2(L(n)) = n \) for any odd number \( n \geq 3 \),
- \( r(L(n)) = n \) for any even number \( n \geq 2 \) and
- \( r(L(n)) = n + 1 \) for any odd number \( n \geq 3 \).

Proof:
- All are more or less easy.
Gossip on Lines

**Theorem:**

We have:

- \( r_2(L(n)) = n - 1 \) for any even number \( n \geq 2 \),
- \( r_2(L(n)) = n \) for any odd number \( n \geq 3 \),
- \( r(L(n)) = n \) for any even number \( n \geq 2 \) and
- \( r(L(n)) = n + 1 \) for any odd number \( n \geq 3 \).

**Proof:**

- All are more or less easy.
Gossip on Lines

Theorem:

We have:
- \( r_2(L(n)) = n - 1 \) for any even number \( n \geq 2 \),
- \( r_2(L(n)) = n \) for any odd number \( n \geq 3 \),
- \( r(L(n)) = n \) for any even number \( n \geq 2 \) and
- \( r(L(n)) = n + 1 \) for any odd number \( n \geq 3 \).

Proof:
- All are more or less easy.
Gossip on arbitrary Trees

Lemma:

For any tree $T$ we have:

- $r(T) = 2 \cdot \text{minb}(T)$
- $r_2(T) = 2 \cdot \text{minb}(T) - 1$

Idea of the proof:

- We have already for any graph $G$: $r(G) \leq 2 \cdot \text{minb}(G)$.
- We have to show: $r(G) \geq 2 \cdot \text{minb}(G)$.
- Let $W = \bigcup_{w \in V} I(v)$ be the total information.
- Let $A$ be any communication algorithm on $T$.
- Let $t$ be the point in time, when some node knows $W$.
- Let $v$ one node, which after $t$ steps know $W$.
- Show: at time $t$ only node $v$ knows $W$. 
Gossip on arbitrary Trees

**Lemma:**

For any tree $T$ we have:

- $r(T) = 2 \cdot \min b(T)$
- $r_2(T) = 2 \cdot \min b(T) - 1$

**Idea of the proof:**

- We have already for any graph $G$: $r(G) \leq 2 \cdot \min b(G)$.
- We have to show: $r(G) \geq 2 \cdot \min b(G)$.
- Let $W = \bigcup_{v \in V} l(v)$ be the total information.
- Let $A$ be any communication algorithm on $T$.
- Let $t$ be the point in time, when some node knows $W$.
- Let $v$ one node, which after $t$ steps know $W$.
- Show: at time $t$ only node $v$ knows $W$. 
Lemma:
For any tree $T$ we have:
- $r(T) = 2 \cdot \min_b(T)$
- $r_2(T) = 2 \cdot \min_b(T) - 1$

Idea of the proof:
- We have already for any graph $G$: $r(G) \leq 2 \cdot \min_b(G)$.
- We have to show: $r(G) \geq 2 \cdot \min_b(G)$.
- Let $W = \bigcup_{w \in V} I(v)$ be the total information.
- Let $A$ be any communication algorithm on $T$.
- Let $t$ be the point in time, when some node knows $W$.
- Let $v$ one node, which after $t$ steps know $W$.
- Show: at time $t$ only node $v$ knows $W$. 
Gossip on arbitrary Trees

Lemma:
For any tree $T$ we have:
- $r(T) = 2 \cdot \minb(T)$
- $r_2(T) = 2 \cdot \minb(T) - 1$

Idea of the proof:
- We have already for any graph $G$: $r(G) \leq 2 \cdot \minb(G)$.
- We have to show: $r(G) \geq 2 \cdot \minb(G)$.
- Let $W = \bigcup_{w \in V} l(v)$ be the total information.
- Let $A$ be any communication algorithm on $T$.
- Let $t$ be the point in time, when some node knows $W$.
- Let $v$ one node, which after $t$ steps know $W$.
- Show: at time $t$ only node $v$ knows $W$. 
Gossip on arbitrary Trees

Lemma:
For any tree \( T \) we have:
- \( r(T) = 2 \cdot \minb(T) \)
- \( r_2(T) = 2 \cdot \minb(T) - 1 \)

Idea of the proof:
- We have already for any graph \( G \): \( r(G) \leq 2 \cdot \minb(G) \).
- We have to show: \( r(G) \geq 2 \cdot \minb(G) \).
- Let \( W = \bigcup_{w \in V} I(v) \) be the total information.
- Let \( A \) be any communication algorithm on \( T \).
- Let \( t \) be the point in time, when some node knows \( W \).
- Let \( v \) one node, which after \( t \) steps know \( W \).
- Show: at time \( t \) only node \( v \) knows \( W \).
Gossip on arbitrary Trees

**Lemma:**

For any tree $T$ we have:

- $r(T) = 2 \cdot \min b(T)$
- $r_2(T) = 2 \cdot \min b(T) - 1$

**Idea of the proof:**

- We have already for any graph $G$: $r(G) \leq 2 \cdot \min b(G)$.
- We have to show: $r(G) \geq 2 \cdot \min b(G)$.
- Let $W = \bigcup_{w \in V} I(v)$ be the total information.
- Let $A$ be any communication algorithm on $T$.
- Let $t$ be the point in time, when some node knows $W$.
- Let $v$ one node, which after $t$ steps know $W$.
- Show: at time $t$ only node $v$ knows $W$. 
Gossip on arbitrary Trees

Lemma:
For any tree $T$ we have:
- $r(T) = 2 \cdot \minb(T)$
- $r_2(T) = 2 \cdot \minb(T) - 1$

Idea of the proof:
- We have already for any graph $G$: $r(G) \leq 2 \cdot \minb(G)$.
- We have to show: $r(G) \geq 2 \cdot \minb(G)$.
- Let $W = \bigcup_{w \in V} I(v)$ be the total information.
- Let $A$ be any communication algorithm on $T$.
- Let $t$ be the point in time, when some node knows $W$.
- Let $v$ one node, which after $t$ steps know $W$.
- Show: at time $t$ only node $v$ knows $W$. 
Gossip on arbitrary Trees

**Lemma:**
For any tree $T$ we have:

- $r(T) = 2 \cdot \min_b(T)$
- $r_2(T) = 2 \cdot \min_b(T) - 1$

**Idea of the proof:**
- We have already for any graph $G$: $r(G) \leq 2 \cdot \min_b(G)$.
- We have to show: $r(G) \geq 2 \cdot \min_b(G)$.
- Let $W = \bigcup_{w \in V} l(v)$ be the total information.
- Let $A$ be any communication algorithm on $T$.
- Let $t$ be the point in time, when some node knows $W$.
- Let $v$ one node, which after $t$ steps know $W$.
- Show: at time $t$ only node $v$ knows $W$. 
Gossip on arbitrary Trees

**Lemma:**
For any tree $T$ we have:
- $r(T) = 2 \cdot \min b(T)$
- $r_2(T) = 2 \cdot \min b(T) - 1$

**Idea of the proof:**
- We have already for any graph $G$: $r(G) \leq 2 \cdot \min b(G)$.
- We have to show: $r(G) \geq 2 \cdot \min b(G)$.
- Let $W = \bigcup_{w \in V} I(v)$ be the total information.
- Let $A$ be any communication algorithm on $T$.
- Let $t$ be the point in time, when some node knows $W$.
- Let $v$ one node, which after $t$ steps know $W$.
- Show: at time $t$ only node $v$ knows $W$. 
Gossip on arbitrary Trees

**Lemma:**

For any tree $T$ we have:

- $r(T) = 2 \cdot \min b(T)$
- $r_2(T) = 2 \cdot \min b(T) - 1$

Idea of the proof:

- We have already for any graph $G$: $r(G) \leq 2 \cdot \min b(G)$.
- We have to show: $r(G) \geq 2 \cdot \min b(G)$.
- Let $W = \bigcup_{w \in V} I(v)$ be the total information.
- Let $A$ be any communication algorithm on $T$.
- Let $t$ be the point in time, when some node knows $W$.
- Let $v$ one node, which after $t$ steps know $W$.
- Show: at time $t$ only node $v$ knows $W$. 
Gossip on arbitrary Trees (Proof I)

- Let $u \neq v$ be an other node which knows $W$ after $t$ steps.
- Let $(u, y_1, y_2, \cdots, y_k, v)$ be the unique path connecting $u$ and $v$.
- If $v$ sends to $y_k$ at time $t$, then $v$ did know $W$ at time $t - 1$.
- So we have to consider the case: $y_k$ sends to $v$ at time $t$:
  - In this case $y_k$ sends $v$ some missing information.
  - $y_k$ knows at time $t - 1$ the full information, which has to be send from $y_k$ to $v$.
  - The information, which has to be send from $v$ to $y_k$, is already send.
  - Then the node $y_k$ know $W$ at time $t - 1$.
- Contradiction, the node $u$ does not exist.
- Thus we have: $t \geq \min b(T) = b(v, T)$.
Gossip on arbitrary Trees (Proof 1)

- Let \( u \neq v \) be an other node which knows \( W \) after \( t \) steps.
- Let \( (u, y_1, y_2, \ldots, y_k, v) \) be the unique path connecting \( u \) and \( v \).
- If \( v \) sends to \( y_k \) at time \( t \), then \( v \) did know \( W \) at time \( t - 1 \).
- So we have to consider the case: \( y_k \) sends to \( v \) at time \( t \):
  - In this case \( y_k \) sends \( v \) some missing information.
  - \( y_k \) knows at time \( t - 1 \) the full information, which has to be send from \( y_k \) to \( v \).
  - The information, which has to be send from \( v \) to \( y_k \), is already send.
  - Then the node \( y_k \) know \( W \) at time \( t - 1 \).

- Contradiction, the node \( u \) does not exist.
- Thus we have: \( t \geq \min_b(T) = b(v, T) \).
Gossip on arbitrary Trees (Proof I)

- Let \( u \neq v \) be an other node which knows \( W \) after \( t \) steps.
- Let \((u, y_1, y_2, \cdots, y_k, v)\) be the unique path connecting \( u \) and \( v \).
- If \( v \) sends to \( y_k \) at time \( t \), then \( v \) did know \( W \) at time \( t - 1 \).
- So we have to consider the case: \( y_k \) sends to \( v \) at time \( t \):
  - In this case \( y_k \) sends \( v \) some missing information.
  - \( y_k \) knows at time \( t - 1 \) the full information, which has to be send from \( y_k \) to \( v \).
  - The information, which has to be send from \( v \) to \( y_k \), is already send.
  - Then the node \( y_k \) know \( W \) at time \( t - 1 \).
- Contradiction, the node \( u \) does not exist.
- Thus we have: \( t \geq \min b(T) = b(v, T) \).
Gossip on arbitrary Trees (Proof I)

- Let $u \neq v$ be an other node which knows $W$ after $t$ steps.
- Let $(u, y_1, y_2, \cdots, y_k, v)$ be the unique path connecting $u$ and $v$.
- If $v$ sends to $y_k$ at time $t$, then $v$ did know $W$ at time $t - 1$.
- So we have to consider the case: $y_k$ sends to $v$ at time $t$:
  - In this case $y_k$ sends $v$ some missing information.
  - $y_k$ knows at time $t - 1$ the full information, which has to be send from $y_k$ to $v$.
  - The information, which has to be send from $v$ to $y_k$, is already send.
  - Then the node $y_k$ know $W$ at time $t - 1$.
- Contradiction, the node $u$ does not exist.
- Thus we have: $t \geq \min b(T) = b(v, T)$.

![Diagram of a tree showing nodes and edges connecting $u$, $y_1$, $y_2$, $y_3$, $y_k$, and $v$.]
Gossip on arbitrary Trees (Proof I)

- Let $u \neq v$ be an other node which knows $W$ after $t$ steps.
- Let $(u, y_1, y_2, \cdots, y_k, v)$ be the unique path connecting $u$ and $v$.
- If $v$ sends to $y_k$ at time $t$, then $v$ did know $W$ at time $t - 1$.
- So we have to consider the case: $y_k$ sends to $v$ at time $t$:
  - In this case $y_k$ sends $v$ some missing information.
  - $y_k$ knows at time $t - 1$ the full information, which has to be send from $y_k$ to $v$.
  - The information, which has to be send from $v$ to $y_k$, is already send.
  - Then the node $y_k$ know $W$ at time $t - 1$.
- Contradiction, the node $u$ does not exist.
- Thus we have: $t \geq \min_b(T) = b(v, T)$. 

\[ u \rightarrow y_1 \rightarrow y_2 \rightarrow y_3 \rightarrow y_k \rightarrow v \]
Gossip on arbitrary Trees (Proof I)

- Let $u \neq v$ be an other node which knows $W$ after $t$ steps.
- Let $(u, y_1, y_2, \ldots, y_k, v)$ be the unique path connecting $u$ and $v$.
- If $v$ sends to $y_k$ at time $t$, then $v$ did know $W$ at time $t - 1$.
- So we have to consider the case: $y_k$ sends to $v$ at time $t$:
  - In this case $y_k$ sends $v$ some missing information.
  - $y_k$ knows at time $t - 1$ the full information, which has to be send from $y_k$ to $v$.
  - The information, which has to be send from $v$ to $y_k$, is already send.
  - Then the node $y_k$ know $W$ at time $t - 1$.
- Contradiction, the node $u$ does not exist.
- Thus we have: $t \geq \min b(T) = b(v, T)$. 

\[ u \rightarrow y_1 \rightarrow y_2 \rightarrow y_3 \rightarrow y_k \rightarrow v \]
Gossip on arbitrary Trees (Proof I)

- Let \( u \neq v \) be an other node which knows \( W \) after \( t \) steps.
- Let \( (u, y_1, y_2, \cdots, y_k, v) \) be the unique path connecting \( u \) and \( v \).
- If \( v \) sends to \( y_k \) at time \( t \), then \( v \) did know \( W \) at time \( t - 1 \).
- So we have to consider the case: \( y_k \) sends to \( v \) at time \( t \):
  - In this case \( y_k \) sends \( v \) some missing information.
  - \( y_k \) knows at time \( t - 1 \) the full information, which has to be send from \( y_k \) to \( v \).
  - The information, which has to be send from \( v \) to \( y_k \), is already send.
  - Then the node \( y_k \) know \( W \) at time \( t - 1 \).
- Contradiction, the node \( u \) does not exist.
- Thus we have: \( t \geq \min b(T) = b(v, T) \).
Gossip on arbitrary Trees (Proof I)

- Let \( u \neq v \) be an other node which knows \( W \) after \( t \) steps.
- Let \((u, y_1, y_2, \ldots, y_k, v)\) be the unique path connecting \( u \) and \( v \).
- If \( v \) sends to \( y_k \) at time \( t \), then \( v \) did know \( W \) at time \( t - 1 \).
- So we have to consider the case: \( y_k \) sends to \( v \) at time \( t \):
  - In this case \( y_k \) sends \( v \) some missing information.
  - \( y_k \) knows at time \( t - 1 \) the full information, which has to be send from \( y_k \) to \( v \).
  - The information, which has to be send from \( v \) to \( y_k \), is already send.
  - Then the node \( y_k \) know \( W \) at time \( t - 1 \).
- Contradiction, the node \( u \) does not exist.
- Thus we have: \( t \geq \min b(T) = b(v, T) \).
Gossip on arbitrary Trees (Proof 1)

- Let $u \neq v$ be an other node which knows $W$ after $t$ steps.
- Let $(u, y_1, y_2, \cdots, y_k, v)$ be the unique path connecting $u$ and $v$.
- If $v$ sends to $y_k$ at time $t$, then $v$ did know $W$ at time $t - 1$.
- So we have to consider the case: $y_k$ sends to $v$ at time $t$:
  - In this case $y_k$ sends $v$ some missing information.
  - $y_k$ knows at time $t - 1$ the full information, which has to be send from $y_k$ to $v$.
  - The information, which has to be send from $v$ to $y_k$, is already send.
  - Then the node $y_k$ know $W$ at time $t - 1$.
- Contradiction, the node $u$ does not exist.
- Thus we have: $t \geq \min_b(T) = b(v, T)$.
Gossip on arbitrary Trees (Proof I)

- Let $u \neq v$ be an other node which knows $W$ after $t$ steps.
- Let $(u, y_1, y_2, \ldots, y_k, v)$ be the unique path connecting $u$ and $v$.
- If $v$ sends to $y_k$ at time $t$, then $v$ did know $W$ at time $t - 1$.
- So we have to consider the case: $y_k$ sends to $v$ at time $t$:
  - In this case $y_k$ sends $v$ some missing information.
  - $y_k$ knows at time $t - 1$ the full information, which has to be send from $y_k$ to $v$.
  - The information, which has to be send from $v$ to $y_k$, is already send.
  - Then the node $y_k$ know $W$ at time $t - 1$.
- Contradiction, the node $u$ does not exist.
- Thus we have: $t \geq \min b(T) = b(v, T)$. 

![Graph showing the nodes and edges connecting them](attachment:image.png)
Gossip on arbitrary Trees (Proof II)

- Consider the situation at node $v$ after round $t$.
- Let w.l.o.g. $v$ be the root of $T$.
- Let $v_1, v_2, \ldots, v_k$ be the successors of $v$.
- Let $T_1, T_2, \ldots, T_k$ be the subtrees with roots $v_1, v_2, \ldots, v_k$.
- In each subtree $T_i$ is some information $w_i$ missing.
- Only the node $v$ knows $\bigcup_{j=1}^{k} w_j$.
- Thus there are $b(v, T)$ steps to be done.
- We finally have $r(T) \geq \min b(T) + b(v, T) \geq 2 \cdot \min b(T)$
Gossip on arbitrary Trees (Proof II)

- Consider the situation at node $v$ after round $t$.
- Let w.l.o.g. $v$ be the root of $T$.
- Let $v_1, v_2, \ldots, v_k$ be the successors of $v$.
- Let $T_1, T_2, \ldots, T_k$ be the subtrees with roots $v_1, v_2, \ldots, v_k$.
- In each subtree $T_i$ is some information $w_i$ missing.
- Only the node $v$ knows $\bigcup_{j=1}^{k} w_j$.
- Thus there are $b(v, T)$ steps to be done.
- We finally have $r(T) \geq \min b(T) + b(v, T) \geq 2 \cdot \min b(T)$
Gossip on arbitrary Trees (Proof II)

- Consider the situation at node $v$ after round $t$.
- Let w.l.o.g. $v$ be the root of $T$.
- Let $v_1, v_2, \ldots, v_k$ be the successors of $v$.
- Let $T_1, T_2, \ldots, T_k$ be the subtrees with roots $v_1, v_2, \ldots, v_k$.
- In each subtree $T_i$ is some information $w_i$ missing.
- Only the node $v$ knows $\bigcup_{j=1}^{k} w_j$.
- Thus there are $b(v, T)$ steps to be done.
- We finally have $r(T) \geq \min b(T) + b(v, T) \geq 2 \cdot \min b(T)$
Gossip on arbitrary Trees (Proof II)

- Consider the situation at node $v$ after round $t$.
- Let w.l.o.g. $v$ be the root of $T$.
- Let $v_1, v_2, \ldots, v_k$ be the successors of $v$.
- Let $T_1, T_2, \ldots, T_k$ be the subtrees with roots $v_1, v_2, \ldots, v_k$.
- In each subtree $T_i$ is some information $w_i$ missing.
- Only the node $v$ knows $\bigcup_{j=1}^{k} w_j$.
- Thus there are $b(v, T)$ steps to be done.
- We finally have $r(T) \geq \min b(T) + b(v, T) \geq 2 \cdot \min b(T)$
Consider the situation at node $v$ after round $t$.
Let w.l.o.g. $v$ be the root of $T$.
Let $v_1, v_2, \ldots, v_k$ be the successors of $v$.
Let $T_1, T_2, \ldots, T_k$ be the subtrees with roots $v_1, v_2, \ldots, v_k$.
In each subtree $T_i$ is some information $w_i$ missing.
 Only the node $v$ knows $\bigcup_{j=1}^{k} w_j$.
Thus there are $b(v, T)$ steps to be done.
We finally have $r(T) \geq \min b(T) + b(v, T) \geq 2 \cdot \min b(T)$
Gossip on arbitrary Trees (Proof II)

- Consider the situation at node $v$ after round $t$.
- Let w.l.o.g. $v$ be the root of $T$.
- Let $v_1, v_2, \cdots, v_k$ be the successors of $v$.
- Let $T_1, T_2, \cdots, T_k$ be the subtrees with roots $v_1, v_2, \cdots, v_k$.
- In each subtree $T_i$ is some information $w_i$ missing.
- **Only the node $v$ knows** $\bigcup_{j=1}^{k} w_j$.
- Thus there are $b(v, T)$ steps to be done.
- We finally have $r(T) \geq \min b(T) + b(v, T) \geq 2 \cdot \min b(T)$. 

![Diagram](attachment:image.png)
Gossip on arbitrary Trees (Proof II)

- Consider the situation at node $v$ after round $t$.
- Let w.l.o.g. $v$ be the root of $T$.
- Let $v_1, v_2, \ldots, v_k$ be the successors of $v$.
- Let $T_1, T_2, \ldots, T_k$ be the subtrees with roots $v_1, v_2, \ldots, v_k$.
- In each subtree $T_i$ is some information $w_i$ missing.
- Only the node $v$ knows $\bigcup_{j=1}^{k} w_j$.
- Thus there are $b(v, T)$ steps to be done.
- We finally have $r(T) \geq \min b(T) + b(v, T) \geq 2 \cdot \min b(T)$
Gossip on arbitrary Trees (Proof II)

- Consider the situation at node $v$ after round $t$.
- Let w.l.o.g. $v$ be the root of $T$.
- Let $v_1, v_2, \ldots, v_k$ be the successors of $v$.
- Let $T_1, T_2, \ldots, T_k$ be the subtrees with roots $v_1, v_2, \ldots, v_k$.
- In each subtree $T_i$ is some information $w_i$ missing.
- Only the node $v$ knows $\cup_{j=1}^{k} w_j$.
- Thus there are $b(v, T)$ steps to be done.
- We finally have $r(T) \geq \min b(T) + b(v, T) \geq 2 \cdot \min b(T)$
Gossip on arbitrary Trees (Proof III)

- Consider the two-way mode: by a similar way we may prove:
- At time $t$ only two neighbours nodes $u$ and $v$ know the total information. We get in the similar way the second statement.
Gossip on arbitrary Trees (Proof III)

Consider the two-way mode: by a similar way we may prove:

- At time $t$ only two neighbours nodes $u$ and $v$ know the total information. We get in the similar way the second statement.
Gossip on arbitrary Trees (Proof III)

- Consider the two-way mode: by a similar way we may prove:
- At time $t$ only two neighbours nodes $u$ and $v$ know the total information. We get in the similar way the second statement.
Gossip on arbitrary Trees (Proof III)

- Consider the two-way mode: by a similar way we may prove:
- At time $t$ only two neighbours nodes $u$ and $v$ know the total information. We get in the similar way the second statement.

![Diagram of a tree with nodes $u$, $v_1$, $v_2$, $v_3$, $v_k$, and $w$, and subtrees $T_1$, $T_2$, $T_3$, $T_k$ connected to $w$.]
Lemma:

For all $m \geq 1$ and $k \geq 2$ we have:

- $r(T_k(m)) = 2 \minb(T_k(m)) = 2 \cdot k \cdot m.$
- $r_2(T_k(m)) = 2 \minb(T_k(m)) - 1 = 2 \cdot k \cdot m - 1.$
Gossip on Cycles

Theorem:

We have:

- \( r_2(C(k)) = \frac{k}{2} \) for even \( k \).
- \( r_2(C(k)) = \lceil \frac{k}{2} \rceil + 1 \) for odd \( k \).

Idea of the proof \((k\ even)\): \([k\ odd: \ an\ easy\ exercise]\)

- Let \( k \) be even.
- \( r_2(C(k)) \geq \frac{k}{2} \) results by the diameter.
- \( r_2(C(k)) \leq \frac{k}{2} \) is true by the following algorithm:
  1. \( \{0, 1\}, \{2, 3\}, \{4, 5\}, \ldots, \{2i, 2i + 1\}, \ldots, \{n - 2, n - 1\} \)
  2. \( \{1, 2\}, \{3, 4\}, \{5, 6\}, \ldots, \{2i - 1, 2i\}, \ldots, \{n - 1, 0\} \)
  3. \( \{0, 1\}, \{2, 3\}, \{4, 5\}, \ldots, \{2i, 2i + 1\}, \ldots, \{n - 2, n - 1\} \)
  4. \( \{1, 2\}, \{3, 4\}, \{5, 6\}, \ldots, \{2i - 1, 2i\}, \ldots, \{n - 1, 0\} \)
  5. \ldots

- Note: After \( i \) rounds knows each node \( 2 \cdot i \) Informationen.
Gossip on Cycles

Theorem:

We have:
- $r_2(C(k)) = k/2$ for even $k$.
- $r_2(C(k)) = \lceil k/2 \rceil + 1$ for odd $k$.

Idea of the proof ($k$ even): [k odd: an easy exercise]

- Let $k$ be even.
- $r_2(C(k)) \geq k/2$ results by the diameter.
- $r_2(C(k)) \leq k/2$ is true by the following algorithm:
  1. $\{\{0,1\}, \{2,3\}, \{4,5\}, \ldots, \{2i, 2i+1\}, \ldots, \{n-2, n-1\}$
  2. $\{\{1,2\}, \{3,4\}, \{5,6\}, \ldots, \{2i-1, 2i\}, \ldots, \{n-1, 0\}$
  3. $\{\{0,1\}, \{2,3\}, \{4,5\}, \ldots, \{2i, 2i+1\}, \ldots, \{n-2, n-1\}$
  4. $\{\{1,2\}, \{3,4\}, \{5,6\}, \ldots, \{2i-1, 2i\}, \ldots, \{n-1, 0\}$
  5. $\ldots$

- Note: After $i$ rounds knows each node $2 \cdot i$ Informationen.
Gossip on Cycles

Theorem:
We have:
- \( r_2(C(k)) = k/2 \) for even \( k \).
- \( r_2(C(k)) = \lceil k/2 \rceil + 1 \) for odd \( k \).

Idea of the proof (\( k \) even): [\( k \) odd: an easy exercise]
- Let \( k \) be even.
- \( r_2(C(k)) \geq k/2 \) results by the diameter.
- \( r_2(C(k)) \leq k/2 \) is true by the following algorithm:
  1. \( \{0, 1\}, \{2, 3\}, \{4, 5\}, \ldots, \{2i, 2i+1\}, \ldots, \{n-2, n-1\} \)
  2. \( \{1, 2\}, \{3, 4\}, \{5, 6\}, \ldots, \{2i-1, 2i\}, \ldots, \{n-1, 0\} \)
  3. \( \{0, 1\}, \{2, 3\}, \{4, 5\}, \ldots, \{2i, 2i+1\}, \ldots, \{n-2, n-1\} \)
  4. \( \{1, 2\}, \{3, 4\}, \{5, 6\}, \ldots, \{2i-1, 2i\}, \ldots, \{n-1, 0\} \)
  5. \( \ldots \)
- Note: After \( i \) rounds knows each node \( 2 \cdot i \) Informationen.
Gossip on Cycles

Theorem:

We have:

- \( r_2(C(k)) = k/2 \) for even \( k \).
- \( r_2(C(k)) = \lceil k/2 \rceil + 1 \) for odd \( k \).

Idea of the proof (\( k \) even): [\( k \) odd: an easy exercise]

- **Let \( k \) be even.**
- \( r_2(C(k)) \geq k/2 \) results by the diameter.
- \( r_2(C(k)) \leq k/2 \) is true by the following algorithm:
  - 1 \{0, 1\}, \{2, 3\}, \{4, 5\}, \ldots, \{2i, 2i + 1\}, \ldots, \{n - 2, n - 1\}
  - 2 \{1, 2\}, \{3, 4\}, \{5, 6\}, \ldots, \{2i - 1, 2i\}, \ldots, \{n - 1, 0\}
  - 3 \{0, 1\}, \{2, 3\}, \{4, 5\}, \ldots, \{2i, 2i + 1\}, \ldots, \{n - 2, n - 1\}
  - 4 \{1, 2\}, \{3, 4\}, \{5, 6\}, \ldots, \{2i - 1, 2i\}, \ldots, \{n - 1, 0\}
  - 5 \ldots
- **Note:** After \( i \) rounds knows each node 2 \( \cdot \) \( i \) Informationen.
Gossip on Cycles

Theorem:

We have:

- \( r_2(C(k)) = k/2 \) for even \( k \).
- \( r_2(C(k)) = \lceil k/2 \rceil + 1 \) for odd \( k \).

Idea of the proof (\( k \) even): [\( k \) odd: an easy exercise]

- Let \( k \) be even.
- \( r_2(C(k)) \geq k/2 \) results by the diameter.
- \( r_2(C(k)) \leq k/2 \) is true by the following algorithm:
  1. \( \{0,1\}, \{2,3\}, \{4,5\}, \ldots, \{2i,2i+1\}, \ldots, \{n-2,n-1\} \)
  2. \( \{1,2\}, \{3,4\}, \{5,6\}, \ldots, \{2i-1,2i\}, \ldots, \{n-1,0\} \)
  3. \( \{0,1\}, \{2,3\}, \{4,5\}, \ldots, \{2i,2i+1\}, \ldots, \{n-2,n-1\} \)
  4. \( \{1,2\}, \{3,4\}, \{5,6\}, \ldots, \{2i-1,2i\}, \ldots, \{n-1,0\} \)
  5. \( \ldots \)

- Note: After \( i \) rounds knows each node \( 2 \cdot i \) Informationen.
Gossip on Cycles

Theorem:
We have:
- $r_2(C(k)) = k/2$ for even $k$.
- $r_2(C(k)) = \lceil k/2 \rceil + 1$ for odd $k$.

Idea of the proof ($k$ even): [ $k$ odd: an easy exercise]
- Let $k$ be even.
- $r_2(C(k)) \geq k/2$ results by the diameter.
- $r_2(C(k)) \leq k/2$ is true by the following algorithm:
  1. $\{\{0, 1\}, \{2, 3\}, \{4, 5\}, \ldots, \{2i, 2i + 1\}, \ldots \}$
  2. $\{\{1, 2\}, \{3, 4\}, \{5, 6\}, \ldots, \{2i - 1, 2i\}, \ldots \}$
  3. $\{\{0, 1\}, \{2, 3\}, \{4, 5\}, \ldots, \{2i, 2i + 1\}, \ldots \}$
  4. $\{\{1, 2\}, \{3, 4\}, \{5, 6\}, \ldots, \{2i - 1, 2i\}, \ldots \}$
  5. $\ldots$

  Note: After $i$ rounds knows each node $2 \cdot i$ Informationen.
Gossip on Cycles

Theorem:

We have:

- \( r_2(C(k)) = \frac{k}{2} \) for even \( k \).
- \( r_2(C(k)) = \lceil \frac{k}{2} \rceil + 1 \) for odd \( k \).

Idea of the proof (\( k \) even): [\( k \) odd: an easy exercise]

- Let \( k \) be even.
- \( r_2(C(k)) \geq \frac{k}{2} \) results by the diameter.
- \( r_2(C(k)) \leq \frac{k}{2} \) is true by the following algorithm:

  1. \( \{0, 1\}, \{2, 3\}, \{4, 5\}, \ldots, \{2i, 2i + 1\}, \ldots, \{n - 2, n - 1\} \)
  2. \( \{1, 2\}, \{3, 4\}, \{5, 6\}, \ldots, \{2i - 1, 2i\}, \ldots, \{n - 1, 0\} \)
  3. \( \{0, 1\}, \{2, 3\}, \{4, 5\}, \ldots, \{2i, 2i + 1\}, \ldots, \{n - 2, n - 1\} \)
  4. \( \{1, 2\}, \{3, 4\}, \{5, 6\}, \ldots, \{2i - 1, 2i\}, \ldots, \{n - 1, 0\} \)
  5. \( \ldots \)

- Note: After \( i \) rounds knows each node \( 2 \cdot i \) Informationen.
Gossip on Cycles

Theorem:

We have:

- $r_2(C(k)) = k/2$ for even $k$.
- $r_2(C(k)) = \lceil k/2 \rceil + 1$ for odd $k$.

Idea of the proof ($k$ even): [k odd: an easy exercise]

- Let $k$ be even.
- $r_2(C(k)) \geq k/2$ results by the diameter.
- $r_2(C(k)) \leq k/2$ is true by the following algorithm:
  - After $i$ rounds knows each node $2 \cdot i$ Informationen.
Gossip on Cycles

Theorem:

We have:
- $r_2(C(k)) = k/2$ for even $k$.
- $r_2(C(k)) = \lceil k/2 \rceil + 1$ for odd $k$.

Idea of the proof ($k$ even): [$k$ odd: an easy exercise]
- Let $k$ be even.
- $r_2(C(k)) \geq k/2$ results by the diameter.
- $r_2(C(k)) \leq k/2$ is true by the following algorithm:
  1. $\{\{0, 1\}, \{2, 3\}, \{4, 5\}, \ldots, \{2i, 2i + 1\}, \ldots, \{n - 2, n - 1\}\$
  2. $\{\{1, 2\}, \{3, 4\}, \{5, 6\}, \ldots, \{2i - 1, 2i\}, \ldots, \{n - 1, 0\}\$
  3. $\{\{0, 1\}, \{2, 3\}, \{4, 5\}, \ldots, \{2i, 2i + 1\}, \ldots, \{n - 2, n - 1\}\$
  4. $\{\{1, 2\}, \{3, 4\}, \{5, 6\}, \ldots, \{2i - 1, 2i\}, \ldots, \{n - 1, 0\}\$
  5. $\ldots$

- Note: After $i$ rounds knows each node $2 \cdot i$ Informationen.
Theorem:

We have:

- $r_2(C(k)) = k/2$ for even $k$.
- $r_2(C(k)) = \lceil k/2 \rceil + 1$ for odd $k$.

Idea of the proof ($k$ even): [k odd: an easy exercise]

- Let $k$ be even.
- $r_2(C(k)) \geq k/2$ results by the diameter.
- $r_2(C(k)) \leq k/2$ is true by the following algorithm:

1. $\{0,1\}, \{2,3\}, \{4,5\}, \ldots, \{2i,2i+1\}, \ldots, \{n-2,n-1\}$
2. $\{1,2\}, \{3,4\}, \{5,6\}, \ldots, \{2i-1,2i\}, \ldots, \{n-1,0\}$
3. $\{0,1\}, \{2,3\}, \{4,5\}, \ldots, \{2i,2i+1\}, \ldots, \{n-2,n-1\}$
4. $\{1,2\}, \{3,4\}, \{5,6\}, \ldots, \{2i-1,2i\}, \ldots, \{n-1,0\}$
5. \ldots

- Note: After $i$ rounds knows each node $2 \cdot i$ Informationen.
Theorem:

We have:

- $r_2(C(k)) = k/2$ for even $k$.
- $r_2(C(k)) = \lceil k/2 \rceil + 1$ for odd $k$.

Idea of the proof ($k$ even): [k odd: an easy exercise]

- Let $k$ be even.
- $r_2(C(k)) \geq k/2$ results by the diameter.
- $r_2(C(k)) \leq k/2$ is true by the following algorithm:

  1. $\{\{0,1\},\{2,3\},\{4,5\},\ldots,\{2i,2i+1\},\ldots,\{n-2,n-1\}\}$
  2. $\{\{1,2\},\{3,4\},\{5,6\},\ldots,\{2i-1,2i\},\ldots,\{n-1,0\}\}$
  3. $\{\{0,1\},\{2,3\},\{4,5\},\ldots,\{2i,2i+1\},\ldots,\{n-2,n-1\}\}$
  4. $\{\{1,2\},\{3,4\},\{5,6\},\ldots,\{2i-1,2i\},\ldots,\{n-1,0\}\}$
  5. $\ldots$

- Note: After $i$ rounds knows each node $2 \cdot i$ Informationen.
Gossip on Cycles

**Theorem:**

We have:
- \( r_2(C(k)) = k/2 \) for even \( k \).
- \( r_2(C(k)) = \lceil k/2 \rceil + 1 \) for odd \( k \).

Idea of the proof (\( k \) even): [\( k \) odd: an easy exercise]

- Let \( k \) be even.
- \( r_2(C(k)) \geq k/2 \) results by the diameter.
- \( r_2(C(k)) \leq k/2 \) is true by the following algorithm:
  - 1
    - \( \{0, 1\} \), \( \{2, 3\} \), \( \{4, 5\} \), \( \cdots \), \( \{2i, 2i + 1\} \), \( \cdots \), \( \{n - 2, n - 1\} \)
  - 2
    - \( \{1, 2\} \), \( \{3, 4\} \), \( \{5, 6\} \), \( \cdots \), \( \{2i - 1, 2i\} \), \( \cdots \), \( \{n - 1, 0\} \)
  - 3
    - \( \{0, 1\} \), \( \{2, 3\} \), \( \{4, 5\} \), \( \cdots \), \( \{2i, 2i + 1\} \), \( \cdots \), \( \{n - 2, n - 1\} \)
  - 4
    - \( \{1, 2\} \), \( \{3, 4\} \), \( \{5, 6\} \), \( \cdots \), \( \{2i - 1, 2i\} \), \( \cdots \), \( \{n - 1, 0\} \)
  - 5
    - \( \cdots \)

- Note: After \( i \) rounds knows each node \( 2 \cdot i \) Informationen.
1-Way Gossip on Cycles (Idea)

- Messages should traverse in both directions.
- Activate each $f(n)$-th node on the cycle.
- This will result in an additional $\Theta(f(n))$ steps.
- During the distribution we get $\Theta\left(\frac{n}{2 \cdot f(n)}\right)$ delays.
- Thus we will choose $f(n) = \Theta(\sqrt{n})$.
- By this idea we may get a lower and upper bound.
1-Way Gossip on Cycles (Idea)

- Messages should traverse in both directions.
- **Activate each** $f(n)$-th node on the cycle.
- This will result in an additional $\Theta(f(n))$ steps.
- During the distribution we get $\Theta\left(\frac{n}{2f(n)}\right)$ delays.
- Thus we will choose $f(n) = \Theta(\sqrt{n})$.
- By this idea we may get a lower and upper bound.
1-Way Gossip on Cycles (Idea)

- Messages should traverse in both directions.
- Activate each $f(n)$-th node on the cycle.
- This will result in an additional $\Theta(f(n))$ steps.
- During the distribution we get $\Theta\left(\frac{n}{2f(n)}\right)$ delays.
- Thus we will choose $f(n) = \Theta(\sqrt{n})$.
- By this idea we may get a lower and upper bound.
1-Way Gossip on Cycles (Idea)

- Messages should traverse in both directions.
- Activate each $f(n)$-th node on the cycle.
- This will result in an additional $\Theta(f(n))$ steps.
- During the distribution we get $\Theta(\frac{n}{2f(n)})$ delays.
- Thus we will choose $f(n) = \Theta(\sqrt{n})$.
- By this idea we may get a lower and upper bound.
1-Way Gossip on Cycles (Idea)

- Messages should traverse in both directions.
- Activate each $f(n)$-th node on the cycle.
- This will result in an additional $\Theta(f(n))$ steps.
- During the distribution we get $\Theta(\frac{n}{2f(n)})$ delays.
- Thus we will choose $f(n) = \Theta(\sqrt{n})$.
- By this idea we may get a lower and upper bound.
1-Way Gossip on Cycles (Idea)

- Messages should traverse in both directions.
- Activate each $f(n)$-th node on the cycle.
- This will result in an additional $\Theta(f(n))$ steps.
- During the distribution we get $\Theta(\frac{n}{2f(n)})$ delays.
- Thus we will choose $f(n) = \Theta(\sqrt{n})$.
- By this idea we may get a lower and upper bound.
Gossip on Cycles (Idea)
Gossip on Cycles (Idea)
Gossip on Cycles (Idea)
Gossip on Cycles (Idea)
Gossip on Cycles (Idea)
Gossip on Cycles (Idea)
Gossip on Cycles (Idea)
Gossip on Cycles (Idea)
Gossip on Cycles (Idea of the algorithm)

- Split the cycle in $\Theta(\sqrt{n})$ blocks $B_i$.
- Within block $B_i$ ($i \in \{1, 2, 3, \ldots, k\}$ with $k \in \Theta(\sqrt{n})$) do the following:
  - Phase 1:
    - The nodes $v_i$ [$u_i$] start a “wave” to the left [right].
    - The messages of $v_i$ and $u_i$ are delayed $\Theta(\sqrt{n})$ times by the other messages.
    - After $n/2 + \Theta(\sqrt{n})$ round know nodes $z_i$ the total information.
  - Phase 2:
    - Each node $z_i$ distribute the total information to $\Theta(\sqrt{n})$ nodes.
- Note: If $n$ is even, we have always a delay of one and the synchronization is easy.
Gossip on Cycles (Idea of the algorithm)

- Split the cycle in $\Theta(\sqrt{n})$ blocks $B_i$.
- Within block $B_i$ ($i \in \{1, 2, 3, \cdots, k\}$ with $k \in \Theta(\sqrt{n})$) do the following:
  - Phase 1:
    - The nodes $v_i$ [$u_i$] start a “wave” to the left [right].
    - The messages of $v_i$ and $u_i$ are delayed $\Theta(\sqrt{n})$ times by the other messages.
    - After $n/2 + \Theta(\sqrt{n})$ round know nodes $z_i$ the total information.
  - Phase 2:
    - Each node $z_i$ distribute the total information to $\Theta(\sqrt{n})$ nodes.
- Note: If $n$ is even, we have always a delay of one and the synchronization is easy.
Gossip on Cycles (Idea of the algorithm)

- Split the cycle in $\Theta(\sqrt{n})$ blocks $B_i$.
- Within block $B_i$ ($i \in \{1, 2, 3, \ldots, k\}$ with $k \in \Theta(\sqrt{n})$) do the following:
  - **Phase 1:**
    - The nodes $v_i$ [$u_i$] start a “wave” to the left [right].
    - The messages of $v_i$ and $u_i$ are delayed $\Theta(\sqrt{n})$ times by the other messages.
    - After $n/2 + \Theta(\sqrt{n})$ round know nodes $z_i$ the total information.
  - **Phase 2:**
    - Each node $z_i$ distribute the total information to $\Theta(\sqrt{n})$ nodes.
- Note: If $n$ is even, we have always a delay of one and the synchronization is easy.
Gossip on Cycles (Idea of the algorithm)

- Split the cycle in $\Theta(\sqrt{n})$ blocks $B_i$.
- Within block $B_i$ ($i \in \{1, 2, 3, \ldots, k\}$ with $k \in \Theta(\sqrt{n})$) do the following:
  - Phase 1:
    - The nodes $v_i$ [$u_i$] start a “wave” to the left [right].
    - The messages of $v_i$ and $u_i$ are delayed $\Theta(\sqrt{n})$ times by the other messages.
    - After $n/2 + \Theta(\sqrt{n})$ round know nodes $z_i$ the total information.
  - Phase 2:
    - Each node $z_i$ distribute the total information to $\Theta(\sqrt{n})$ nodes.
- Note: If $n$ is even, we have always a delay of one and the synchronization is easy.
Gossip on Cycles (Idea of the algorithm)

- Split the cycle in $\Theta(\sqrt{n})$ blocks $B_i$.
- Within block $B_i$ ($i \in \{1, 2, 3, \ldots, k\}$ with $k \in \Theta(\sqrt{n})$) do the following:
  - Phase 1:
    - The nodes $v_i$ [$u_i$] start a “wave” to the left [right].
    - The messages of $v_i$ and $u_i$ are delayed $\Theta(\sqrt{n})$ times by the other messages.
    - After $n/2 + \Theta(\sqrt{n})$ round know nodes $z_i$ the total information.
  - Phase 2:
    - Each node $z_i$ distribute the total information to $\Theta(\sqrt{n})$ nodes.
- Note: If $n$ is even, we have always a delay of one and the synchronization is easy.
Gossip on Cycles (Idea of the algorithm)

- Split the cycle in $\Theta(\sqrt{n})$ blocks $B_i$.
- Within block $B_i$ ($i \in \{1, 2, 3, \cdots, k\}$ with $k \in \Theta(\sqrt{n})$) do the following:
  - Phase 1:
    - The nodes $v_i$ [$u_i$] start a “wave” to the left [right].
    - The messages of $v_i$ and $u_i$ are delayed $\Theta(\sqrt{n})$ times by the other messages.
    - After $n/2 + \Theta(\sqrt{n})$ round know nodes $z_i$ the total information.
  - Phase 2:
    - Each node $z_i$ distribute the total information to $\Theta(\sqrt{n})$ nodes.
- Note: If $n$ is even, we have always a delay of one and the synchronization is easy.
Gossip on Cycles (Idea of the algorithm)

- Split the cycle in $\Theta(\sqrt{n})$ blocks $B_i$.
- Within block $B_i$ ($i \in \{1, 2, 3, \ldots, k\}$ with $k \in \Theta(\sqrt{n})$) do the following:
  - Phase 1:
    - The nodes $v_i [u_i]$ start a “wave” to the left [right].
    - The messages of $v_i$ and $u_i$ are delayed $\Theta(\sqrt{n})$ times by the other messages.
    - After $n/2 + \Theta(\sqrt{n})$ round know nodes $z_i$ the total information.
  - Phase 2:
    - Each node $z_i$ distribute the total information to $\Theta(\sqrt{n})$ nodes.
- Note: If $n$ is even, we have always a delay of one and the synchronization is easy.
Gossip on Cycles (Idea of the algorithm)

- Split the cycle in $\Theta(\sqrt{n})$ blocks $B_i$.
- Within block $B_i$ ($i \in \{1, 2, 3, \ldots, k\}$ with $k \in \Theta(\sqrt{n})$) do the following:
  - Phase 1:
    - The nodes $v_i$ [or $u_i$] start a “wave” to the left [or right].
    - The messages of $v_i$ and $u_i$ are delayed $\Theta(\sqrt{n})$ times by the other messages.
    - After $n/2 + \Theta(\sqrt{n})$ round know nodes $z_i$ the total information.
  - Phase 2:
    - Each node $z_i$ distribute the total information to $\Theta(\sqrt{n})$ nodes.

Note: If $n$ is even, we have always a delay of one and the synchronization is easy.

![Diagram of the cycle with nodes and edges labeled for the gossip algorithm description.]
Gossip on Cycles (Idea of the algorithm)

- Split the cycle in $\Theta(\sqrt{n})$ blocks $B_i$.
- Within block $B_i$ ($i \in \{1, 2, 3, \ldots, k\}$ with $k \in \Theta(\sqrt{n})$) do the following:
  - Phase 1:
    - The nodes $v_i$ [$u_i$] start a “wave” to the left [right].
    - The messages of $v_i$ and $u_i$ are delayed $\Theta(\sqrt{n})$ times by the other messages.
    - After $n/2 + \Theta(\sqrt{n})$ round know nodes $z_i$ the total information.
  - Phase 2:
    - Each node $z_i$ distribute the total information to $\Theta(\sqrt{n})$ nodes.
- Note: If $n$ is even, we have always a delay of one and the synchronization is easy.
Gossip on Cycles (Idea)

Theorem:

We have:

- \( r(C(n)) \leq \frac{n}{2} + \sqrt{2n} - 1 \) for even \( n \).
- \( r(C(n)) \leq \lceil \frac{n}{2} \rceil + \lceil 2 \cdot \sqrt{\lceil \frac{n}{2} \rceil} \rceil - 1 \) for odd \( n \).
- \( r(C(n)) \geq \frac{n}{2} + \sqrt{2n} - 1 \) for even \( n \).
- \( r(C(n)) \geq \lceil \frac{n}{2} \rceil + \lceil \sqrt{2n} - 1/2 \rceil - 1 \) for odd \( n \).

Proof: See literature.
Gossip on the Hypercube

Theorem:
For all $m \in \mathbb{N}$ we have: $r_2(HQ(m)) = m$

Proof:
- The lower bound is the diameter.
- Upper bound by the following algorithm:
  
  ```
  for $i = 1$ to $m$ do 
    for all $a_1, a_2, \ldots, a_{m-1} \in \{0, 1\}$ do in parallel 
      $a_1a_2 \cdots a_{i-1}0a_ia_{i+1} \cdots a_{m-1}$ sends to 
      $a_1a_2 \cdots a_{i-1}1a_ia_{i+1} \cdots a_{m-1}$
  ```

Corollary:
For all $m \in \mathbb{N}$ we have: $r_2(K(2^m)) = m$
Gossip on the Hypercube

**Theorem:**
For all $m \in \mathbb{N}$ we have: $r_2(HQ(m)) = m$

**Proof:**
- The lower bound is the diameter.
- Upper bound by the following algorithm:
  ```
  for $i = 1$ to $m$ do
    for all $a_1, a_2, \ldots, a_{m-1} \in \{0, 1\}$ do in parallel
      $a_1a_2\cdots a_{i-1}0a_ia_{i+1}\cdots a_{m-1}$ sends to
      $a_1a_2\cdots a_{i-1}1a_ia_{i+1}\cdots a_{m-1}$
  ```

**Corollary:**
For all $m \in \mathbb{N}$ we have: $r_2(K(2^m)) = m$
Gossip on the Hypercube

**Theorem:**

For all $m \in \mathbb{N}$ we have: $r_2(HQ(m)) = m$

**Proof:**

- The lower bound is the diameter.
- Upper bound by the following algorithm:
  
  ```
  for $i = 1$ to $m$ do 
  for all $a_1, a_2, \ldots, a_{m-1} \in \{0, 1\}$ do in parallel 
  $a_1 a_2 \cdots a_{i-1} 0 a_i a_{i+1} \cdots a_{m-1}$ sends to 
  $a_1 a_2 \cdots a_{i-1} 1 a_i a_{i+1} \cdots a_{m-1}$
  ```

**Corollary:**

For all $m \in \mathbb{N}$ we have: $r_2(K(2^m)) = m$
Gossip on the Hypercube

Theorem:
For all \( m \in \mathbb{N} \) we have: \( r_2(HQ(m)) = m \)

Proof:
- The lower bound is the diameter.
- Upper bound by the following algorithm:
  
  \[
  \text{for } i = 1 \text{ to } m \text{ do }
  \]
  
  \[
  \text{for all } a_1, a_2, \ldots, a_{m-1} \in \{0, 1\} \text{ do in parallel }
  \]
  
  \[
  a_1a_2\cdots a_{i-1}0a_ia_{i+1}\cdots a_{m-1} \text{ sends to }
  \]
  
  \[
  a_1a_2\cdots a_{i-1}1a_ia_{i+1}\cdots a_{m-1}
  \]

Corollary:
For all \( m \in \mathbb{N} \) we have: \( r_2(K(2^m)) = m \)
Consider one-way mode:

- Start with the first phase of the gossip-algorithm for cycles on all cycles.
- Then each $\Theta(\sqrt{n})$-th node on each cycle knows the total information of its cycles.
- In $\Theta(\sqrt{n})$ waves distribute this information down and between the cycles.
- After $\Theta(n)$ steps knows each $\Theta(\sqrt{n})$-th node of each cycle the total information.
- The final part is the second phase of the gossip-algorithm of cycles on all cycles.
- All nodes know now the total information.
Consider one-way mode:

- Start with the first phase of the gossip-algorithm for cycles on all cycles.
- Then each $\Theta(\sqrt{n})$-th node on each cycle knows the total information of its cycles.
- In $\Theta(\sqrt{n})$ waves distribute this information down and between the cycles.
- After $\Theta(n)$ steps knows each $\Theta(\sqrt{n})$-th node of each cycle the total information.
- The final part is the second phase of the gossip-algorithm of cycles on all cycles.
- All nodes know now the total information.
Consider one-way mode:

- Start with the first phase of the gossip-algorithm for cycles on all cycles.
- Then each $\Theta(\sqrt{n})$-th node on each cycle knows the total information of its cycles.
- In $\Theta(\sqrt{n})$ waves distribute this information down and between the cycles.
- After $\Theta(n)$ steps knows each $\Theta(\sqrt{n})$-th node of each cycle the total information.
- The final part is the second phase of the gossip-algorithm of cycles on all cycles.
- All nodes know now the total information.
Consider one-way mode:

- Start with the first phase of the gossip-algorithm for cycles on all cycles.
- Then each $\Theta(\sqrt{n})$-th node on each cycle knows the total information of its cycles.
- In $\Theta(\sqrt{n})$ waves distribute this information down and between the cycles.
- After $\Theta(n)$ steps knows each $\Theta(\sqrt{n})$-th node of each cycle the total information.
- The final part is the second phase of the gossip-algorithm of cycles on all cycles.
- All nodes know now the total information.
Consider one-way mode:

- Start with the first phase of the gossip-algorithm for cycles on all cycles.
- Then each $\Theta(\sqrt{n})$-th node on each cycle knows the total information of its cycles.
- In $\Theta(\sqrt{n})$ waves distribute this information down and between the cycles.
- After $\Theta(n)$ steps knows each $\Theta(\sqrt{n})$-th node of each cycle the total information.
- The final part is the second phase of the gossip-algorithm of cycles on all cycles.
- All nodes know now the total information.
Consider one-way mode:

- Start with the first phase of the gossip-algorithm for cycles on all cycles.
- Then each $\Theta(\sqrt{n})$-th node on each cycle knows the total information of its cycles.
- In $\Theta(\sqrt{n})$ waves distribute this information down and between the cycles.
- After $\Theta(n)$ steps knows each $\Theta(\sqrt{n})$-th node of each cycle the total information.
- The final part is the second phase of the gossip-algorithm of cycles on all cycles.
- All nodes know now the total information.
Consider one-way mode:
- Start with the first phase of the gossip-algorithm for cycles on all cycles.
- Then each $\Theta(\sqrt{n})$-th node on each cycle knows the total information of its cycles.
- In $\Theta(\sqrt{n})$ waves distribute this information down and between the cycles.
- After $\Theta(n)$ steps knows each $\Theta(\sqrt{n})$-th node of each cycle the total information.
- The final part is the second phase of the gossip-algorithm of cycles on all cycles.
- All nodes know now the total information.
Consider two-way mode:

- Start with the gossip algorithm for cycles on all cycles.
- Each node of the cycle knows now the total information of its cycle.
- In $\Theta(n/2)$ waves distribute this information down and between the cycles.
- After $\Theta(n)$ steps knows each node the total information.
Consider two-way mode:

- Start with the gossip algorithm for cycles on all cycles.
- Each node of the cycle knows now the total information of its cycle.
- In $\Theta(n/2)$ waves distribute this information down and between the cycles.
- After $\Theta(n)$ steps knows each node the total information.
Consider two-way mode:

- Start with the gossip algorithm for cycles on all cycles.
- Each node of the cycle knows now the total information of its cycle.
- In $\Theta(n/2)$ waves distribute this information down and between the cycles.
- After $\Theta(n)$ steps knows each node the total information.
Consider two-way mode:

- Start with the gossip algorithm for cycles on all cycles.
- Each node of the cycle knows now the total information of its cycle.
- In $\Theta(n/2)$ waves distribute this information down and between the cycles.
- After $\Theta(n)$ steps knows each node the total information.
Consider two-way mode:
- Start with the gossip algorithm for cycles on all cycles.
- Each node of the cycle knows now the total information of its cycle.
- In $\Theta(n/2)$ waves distribute this information down and between the cycles.
- After $\Theta(n)$ steps knows each node the total information.
Theorem:

Let $k \geq 3$, then we have:

- $r(\text{CCC}(k)) \leq r(\text{C}(k)) + 3k - 1 \leq \left\lceil \frac{7k}{2} \right\rceil + \left\lceil 2\sqrt{\left\lceil \frac{k}{2} \right\rceil} \right\rceil - 2$.

- $r(\text{BF}(k)) \leq r(\text{C}(k)) + 2k \leq \left\lceil \frac{5k}{2} \right\rceil + \left\lceil 2\sqrt{\left\lceil \frac{k}{2} \right\rceil} \right\rceil - 1$.

- $r_2(\text{CCC}(k)) \leq \frac{k}{2} + 2k = 5 \cdot \left\lceil \frac{k}{2} \right\rceil$ for even $k$.

- $r_2(\text{CCC}(k)) \leq \left\lceil \frac{k}{2} \right\rceil + 2k + 2 = 5 \cdot \left\lceil \frac{k}{2} \right\rceil$ for odd $k$.

- $r_2(\text{BF}(k)) \leq \frac{k}{2} + 2k = 5 \cdot \left\lceil \frac{k}{2} \right\rceil$ for even $k$.

- $r_2(\text{BF}(k)) \leq \left\lceil \frac{k}{2} \right\rceil + 2k + 2 = 5 \cdot \left\lceil \frac{k}{2} \right\rceil$ for odd $k$. 
Theorem:

Let \( k \geq 3 \), then we have:

1. \( r(\text{CCC}(k)) \leq r(\text{C}(k)) + 3k - 1 \leq \left\lceil \frac{7k}{2} \right\rceil + \left\lfloor 2\sqrt{\left\lfloor \frac{k}{2} \right\rfloor} \right\rfloor - 2 \).
2. \( r(\text{BF}(k)) \leq r(\text{C}(k)) + 2k \leq \left\lceil \frac{5k}{2} \right\rceil + \left\lfloor 2\sqrt{\left\lfloor \frac{k}{2} \right\rfloor} \right\rfloor - 1 \).
3. \( r_2(\text{CCC}(k)) \leq \frac{k}{2} + 2k = 5 \cdot \left\lceil \frac{k}{2} \right\rceil \) for even \( k \).
4. \( r_2(\text{CCC}(k)) \leq \left\lceil \frac{k}{2} \right\rceil + 2k + 2 = 5 \cdot \left\lceil \frac{k}{2} \right\rceil \) for odd \( k \).
5. \( r_2(\text{BF}(k)) \leq \frac{k}{2} + 2k = 5 \cdot \left\lceil \frac{k}{2} \right\rceil \) for even \( k \).
6. \( r_2(\text{BF}(k)) \leq \left\lceil \frac{k}{2} \right\rceil + 2k + 2 = 5 \cdot \left\lceil \frac{k}{2} \right\rceil \) for odd \( k \).
Theorem:

Let \( k \geq 3 \), then we have:

- \( r(\text{CCC}(k)) \leq r(C(k)) + 3k - 1 \leq \left\lceil \frac{7k}{2} \right\rceil + \left\lceil 2\sqrt{\left\lceil \frac{k}{2} \right\rceil} \right\rceil - 2. \)

- \( r(\text{BF}(k)) \leq r(C(k)) + 2k \leq \left\lceil \frac{5k}{2} \right\rceil + \left\lceil 2\sqrt{\left\lceil \frac{k}{2} \right\rceil} \right\rceil - 1. \)

- \( r_2(\text{CCC}(k)) \leq \frac{k}{2} + 2k = 5 \cdot \left\lceil \frac{k}{2} \right\rceil \) for even \( k \).

- \( r_2(\text{CCC}(k)) \leq \left\lceil \frac{k}{2} \right\rceil + 2k + 2 = 5 \cdot \left\lceil \frac{k}{2} \right\rceil \) for odd \( k \).

- \( r_2(\text{BF}(k)) \leq \frac{k}{2} + 2k = 5 \cdot \left\lceil \frac{k}{2} \right\rceil \) for even \( k \).

- \( r_2(\text{BF}(k)) \leq \left\lceil \frac{k}{2} \right\rceil + 2k + 2 = 5 \cdot \left\lceil \frac{k}{2} \right\rceil \) for odd \( k \).
Theorem:

Let $k \geq 3$, then we have:

- $r(\text{CCC}(k)) \leq r(C(k)) + 3k - 1 \leq \left\lceil \frac{7k}{2} \right\rceil + \left\lceil 2\sqrt{\left\lceil \frac{k}{2} \right\rceil} \right\rceil - 2$.

- $r(\text{BF}(k)) \leq r(C(k)) + 2k \leq \left\lceil \frac{5k}{2} \right\rceil + \left\lceil 2\sqrt{\left\lceil \frac{k}{2} \right\rceil} \right\rceil - 1$.

- $r_2(\text{CCC}(k)) \leq \frac{k}{2} + 2k = 5 \cdot \left\lceil \frac{k}{2} \right\rceil$ for even $k$.

- $r_2(\text{CCC}(k)) \leq \left\lceil \frac{k}{2} \right\rceil + 2k + 2 = 5 \cdot \left\lceil \frac{k}{2} \right\rceil$ for odd $k$.

- $r_2(\text{BF}(k)) \leq \frac{k}{2} + 2k = 5 \cdot \left\lceil \frac{k}{2} \right\rceil$ for even $k$.

- $r_2(\text{BF}(k)) \leq \left\lceil \frac{k}{2} \right\rceil + 2k + 2 = 5 \cdot \left\lceil \frac{k}{2} \right\rceil$ for odd $k$. 
Theorem:

Let $k \geq 3$, then we have:

- $r(\text{CCC}(k)) \leq r(\text{C}(k)) + 3k - 1 \leq \lceil \frac{7k}{2} \rceil + 2\sqrt{\lceil \frac{k}{2} \rceil} - 2$.
- $r(\text{BF}(k)) \leq r(\text{C}(k)) + 2k \leq \lceil \frac{5k}{2} \rceil + 2\sqrt{\lceil \frac{k}{2} \rceil} - 1$.
- $r_2(\text{CCC}(k)) \leq \frac{k}{2} + 2k = 5 \cdot \lceil \frac{k}{2} \rceil$ for even $k$.
- $r_2(\text{CCC}(k)) \leq \lceil \frac{k}{2} \rceil + 2k + 2 = 5 \cdot \lceil \frac{k}{2} \rceil$ for odd $k$.
- $r_2(\text{BF}(k)) \leq \frac{k}{2} + 2k = 5 \cdot \lceil \frac{k}{2} \rceil$ for even $k$.
- $r_2(\text{BF}(k)) \leq \lceil \frac{k}{2} \rceil + 2k + 2 = 5 \cdot \lceil \frac{k}{2} \rceil$ for odd $k$. 
Theorem:

Let \( k \geq 3 \), then we have:

- \( r(\text{CCC}(k)) \leq r(\text{C}(k)) + 3k - 1 \leq \left\lceil \frac{7k}{2} \right\rceil + \left\lceil 2\sqrt{\left\lceil \frac{k}{2} \right\rceil} \right\rceil - 2. \)
- \( r(\text{BF}(k)) \leq r(\text{C}(k)) + 2k \leq \left\lceil \frac{5k}{2} \right\rceil + \left\lceil 2\sqrt{\left\lceil \frac{k}{2} \right\rceil} \right\rceil - 1. \)
- \( r_2(\text{CCC}(k)) \leq \frac{k}{2} + 2k = 5 \cdot \left\lceil \frac{k}{2} \right\rceil \) for even \( k \).
- \( r_2(\text{CCC}(k)) \leq \left\lceil \frac{k}{2} \right\rceil + 2k + 2 = 5 \cdot \left\lceil \frac{k}{2} \right\rceil \) for odd \( k \).
- \( r_2(\text{BF}(k)) \leq \frac{k}{2} + 2k = 5 \cdot \left\lceil \frac{k}{2} \right\rceil \) for even \( k \).
- \( r_2(\text{BF}(k)) \leq \left\lceil \frac{k}{2} \right\rceil + 2k + 2 = 5 \cdot \left\lceil \frac{k}{2} \right\rceil \) for odd \( k \).
Gossip on Graphs with $2 \cdot m$ Nodes (0. Idea)
Gossip on Graphs with $2 \cdot m$ Nodes (0. Idea)
Gossip on Graphs with $2 \cdot m$ Nodes (0. Idea)
Gossip on Graphs with $2 \cdot m$ Nodes (0. Idea)
Gossip on Graphs with $2 \cdot m$ Nodes (0. Idea)
Gossip on Graphs with $2 \cdot m$ Nodes (0. Idea)
Gossip on Graphs with $2 \cdot m$ Nodes (0. Idea)
Gossip on Graphs with $2 \cdot m$ Nodes (0. Idea)
Gossip on Graphs with $2 \cdot m$ Nodes (1. Idea)
Gossip on Graphs with $2 \cdot m$ Nodes (1. Idea)

Implication: For all $m \in \mathbb{N}$ we have:
$$r_2(K_{2^m}) = m$$

For all $m \in \mathbb{N}$ we have:
$$r_2(K_m) \leq \lceil \log m \rceil + 1$$
Gossip on Graphs with $2 \cdot m$ Nodes (1. Idea)

Implication:
For all $m \in \mathbb{N}$ we have:
$$r_2(K(2^m)) = m$$

For all $m \in \mathbb{N}$ we have:
$$r_2(K(m)) \leq \lceil \log m \rceil + 1$$
Gossip on Graphs with $2 \cdot m$ Nodes (1. Idea)

Implication:
For all $m \in \mathbb{N}$ we have:
$$r_2(K_{2^m}) = m$$

For all $m \in \mathbb{N}$ we have:
$$r_2(K_m) \leq \lceil \log_2 m \rceil + 1$$
Gossip on Graphs with $2 \cdot m$ Nodes (1. Idea)

Implication:

For all $m \in \mathbb{N}$ we have:

$r_2(K_{2^m}) = m$

For all $m \in \mathbb{N}$ we have:

$r_2(K_{m}) \leq \lceil \log m \rceil + 1$
Gossip on Graphs with $2 \cdot m$ Nodes (1. Idea)

Implication:

For all $m \in \mathbb{N}$ we have:

$r_{2}(K(2^{m})) = m$

For all $m \in \mathbb{N}$ we have:

$r_{2}(K(m)) \leq \left\lceil \log m \right\rceil + 1$
Gossip on Graphs with $2 \cdot m$ Nodes (1. Idea)

Implication:
For all $m \in \mathbb{N}$ we have:
$$r_2(K_{2^m}) = m$$

For all $m \in \mathbb{N}$ we have:
$$r_2(K_m) \leq \lceil \log m \rceil + 1$$
Gossip on Graphs with $2 \cdot m$ Nodes (1. Idea)

Implication:
For all $m \in \mathbb{N}$ we have:
$$r_2(K_{2^m}) = m$$

For all $m \in \mathbb{N}$ we have:
$$r_2(K_m) \leq \lceil \log m \rceil + 1$$
Gossip on Graphs with $2 \cdot m$ Nodes (1. Idea)

Implication:
For all $m \in \mathbb{N}$ we have:
$$r_2(K(2^m)) = m$$

For all $m \in \mathbb{N}$ we have:
$$r_2(K(m)) \leq \lceil \log_2 m \rceil + 1$$
Gossip on Graphs with $2 \cdot m$ Nodes (1. Idea)

Implication:

For all $m \in \mathbb{N}$ we have:

$$r_2(K(2^m)) = m$$

For all $m \in \mathbb{N}$ we have:

$$r_2(K(m)) \leq \lceil \log_2 m \rceil + 1$$
Gossip on Graphs with $2 \cdot m$ Nodes (1. Idea)

Implication:

- For all $m \in \mathbb{N}$ we have: $r_2(K(2^m)) = m$
- For all $m \in \mathbb{N}$ we have: $r_2(K(m)) \leq \lceil \log m \rceil + 1$
Gossip on Graphs with $2 \cdot m$ Nodes (2. Idea)

- Too many nodes where inactive for too long time.
- These nodes could not double their information.
- Idea: Try to double the information of any node.
- Detailed idea: In each step each node has an “interval” of information.
- To make the doubling easy split the nodes into two groups.
- Both groups should be the same size.
- In the first step pairs of node from each group share their information.
Gossip on Graphs with $2 \cdot m$ Nodes (2. Idea)

- Too many nodes where inactive for too long time.
- These nodes could not double their information.
- Idea: Try to double the information of any node.
- Detailed idea: In each step each node has an “interval” of information.
- To make the doubling easy split the nodes into two groups.
- Both groups should be the same size.
- In the first step pairs of node from each group share their information.
Gossip on Graphs with $2 \cdot m$ Nodes (2. Idea)

- Too many nodes where inactive for too long time.
- These nodes could not double their information.
- Idea: Try to double the information of any node.
- Detailed idea: In each step each node has an “interval” of information.
- To make the doubling easy split the nodes into two groups.
- Both groups should be the same size.
- In the first step pairs of node from each group share their information.
Gossip on Graphs with $2 \cdot m$ Nodes (2. Idea)

- Too many nodes where inactive for too long time.
- These nodes could not double their information.
- Idea: Try to double the information of any node.
- Detailed idea: In each step each node has an “interval” of information.
- To make the doubling easy split the nodes into two groups.
- Both groups should be the same size.
- In the first step pairs of node from each group share their information.
Gossip on Graphs with $2 \cdot m$ Nodes (2. Idea)

- Too many nodes where inactive for too long time.
- These nodes could not double their information.
- Idea: Try to double the information of any node.
- Detailed idea: In each step each node has an “interval” of information.
- To make the doubling easy split the nodes into two groups.
- Both groups should be the same size.
- In the first step pairs of node from each group share their information.
Gossip on Graphs with $2 \cdot m$ Nodes (2. Idea)

- Too many nodes where inactive for too long time.
- These nodes could not double their information.
- Idea: Try to double the information of any node.
- Detailed idea: In each step each node has an “interval” of information.
- To make the doubling easy split the nodes into two groups.
- Both groups should be the same size.
- In the first step pairs of node from each group share their information.
Gossip on Graphs with $2 \cdot m$ Nodes (2. Idea)

- Too many nodes where inactive for too long time.
- These nodes could not double their information.
- Idea: Try to double the information of any node.
- Detailed idea: In each step each node has an “interval” of information.
- To make the doubling easy split the nodes into two groups.
- Both groups should be the same size.
- In the first step pairs of node from each group share their information.
Gossip on Graphs with $2 \cdot m$ Nodes (2. Idea)
Gossip on Graphs with $2 \cdot m$ Nodes (2. Idea)
Gossip on Graphs with $2 \cdot m$ Nodes (2. Idea)
Gossip on Graphs with $2 \cdot m$ Nodes (2. Idea)
Gossip on Graphs with $2 \cdot m$ Nodes (2. Idea)
Gossip on Graphs with $2 \cdot m$ Nodes (2. Idea)
Gossip on Graphs with $2 \cdot m$ Nodes (2. Idea)
Gossip on Graphs with $2 \cdot m$ Nodes (2. Idea)
Gossip on Graphs with $2 \cdot m$ Nodes (2. Idea)
Gossip on Graphs with $2 \cdot m$ Nodes (2. Idea)
Gossip on Graphs with $2 \cdot m$ Nodes

**Theorem:**
For all $m \in \mathbb{N}$ we have: $r_2(K(2m)) = \lceil \log 2m \rceil$.

**Proof:** Split the nodes in groups $Q[i]$ and $R[i]$ ($0 \leq i \leq m - 1$).

- **algorithm:**
  
  for all $i \in \{0, \ldots, m - 1\}$ do in parallel
  Exchange the information between $Q[i]$ and $R[i]$.

  for $t = 1$ to $\lceil \log_2 m \rceil$ do
    
    for all $i \in \{0, \ldots, m - 1\}$ do in parallel
    Exchange the information between $Q[i]$ and $R[(i + 2^{t-1}) \mod m]$.

- **Invariant:**
  
  Let $\alpha[i]$ be the information of $Q[i]$ and $R[i]$ after their initial exchange.

  After round $t$ know nodes $Q[i]$ and $R[(i + 2^{t-1}) \mod m]$: $\cup_{0 \leq j \leq 2^{t-1}} \alpha[(i + j) \mod m]$.

  The invariant is easy to be shown.
Gossip on Graphs with $2 \cdot m$ Nodes

**Theorem:**
For all $m \in \mathbb{N}$ we have: $r_2(K(2m)) = \lceil \log 2m \rceil$

**Proof:** Split the nodes in groups $Q[i]$ and $R[i]$ ($0 \leq i \leq m - 1$).

- **algorithm:**
  
  for all $i \in \{0, \cdots, m - 1\}$ do in parallel
  
  Exchange the information between $Q[i]$ and $R[i]$

  for $t = 1$ to $\lceil \log_2 m \rceil$ do
    
    for all $i \in \{0, \ldots, m - 1\}$ do in parallel
    
    Exchange the information between $Q[i]$ and $R[(i + 2^{t-1}) \mod m]$

- **Invariant:**
  
  Let $\alpha[i]$ be the information of $Q[i]$ and $R[i]$ after their initial exchange.

  After round $t$ know nodes $Q[i]$ and $R[(i + 2^{t-1}) \mod m]$: $\bigcup_{0 \leq j \leq 2^{t-1}} \alpha[(i + j) \mod m]$

  The invariant is easy to be shown.
Gossip on Graphs with \(2 \cdot m\) Nodes

**Theorem:**

For all \(m \in \mathbb{N}\) we have: 
\[r_2(K(2m)) = \lceil \log 2m \rceil\]

**Proof:** Split the nodes in groups \(Q[i]\) and \(R[i]\) \((0 \leq i \leq m - 1)\).

- **algorithm:**
  
  ```
  for all \(i \in \{0, \cdots, m-1\}\) do in parallel
  Exchange the information between \(Q[i]\) and \(R[i]\)
  for \(t = 1\) to \(\lceil \log_2 m \rceil\) do
    for all \(i \in \{0, \cdots, m-1\}\) do in parallel
      Exchange the information between \(Q[i]\) and \(R[(i + 2^{t-1}) \mod m]\)
  ```

- **Invariant:**
  
  - Let \(\alpha[i]\) be the information of \(Q[i]\) and \(R[i]\) after their initial exchange.
  - After round \(t\) know nodes \(Q[i]\) and \(R[(i + 2^{t-1}) \mod m]\):
    \[\bigcup_{0 \leq j \leq 2^{t-1}} \alpha[(i + j) \mod m]\]
  - The invariant is easy to be shown.
Gossip on Graphs with $2 \cdot m$ Nodes

Theorem:
For all $m \in \mathbb{N}$ we have: $r_2(K(2m)) = \lceil \log 2m \rceil$

Proof: Split the nodes in groups $Q[i]$ and $R[i]$ ($0 \leq i \leq m - 1$).

- algorithm:
  ```plaintext
  for all $i \in \{0, \ldots, m - 1\}$ do in parallel
    Exchange the information between $Q[i]$ and $R[i]$
  for $t = 1$ to $\lceil \log_2 m \rceil$ do
    for all $i \in \{0, \ldots, m - 1\}$ do in parallel
      Exchange the information between $Q[i]$ and $R[(i + 2^{t-1}) \mod m]$
  ```

- Invariant:
  - Let $\alpha[i]$ be the information of $Q[i]$ and $R[i]$ after their initial exchange.
  - After round $t$ know nodes $Q[i]$ and $R[(i + 2^{t-1}) \mod m]$: $\bigcup_{0 \leq j \leq 2^{t-1}} \alpha[(i + j) \mod m]$
  - The invariant is easy to be shown.
Gossip on Graphs with $2 \cdot m$ Nodes

**Theorem:**
For all $m \in \mathbb{N}$ we have: $r_2(K(2m)) = \lceil \log 2m \rceil$

**Proof:** Split the nodes in groups $Q[i]$ and $R[i]$ ($0 \leq i \leq m - 1$).

- **algorithm:**
  ```
  for all $i \in \{0, \ldots, m - 1\}$ do in parallel
    Exchange the information between $Q[i]$ and $R[i]$
  for $t = 1$ to $\lceil \log_2 m \rceil$ do
    for all $i \in \{0, \ldots, m - 1\}$ do in parallel
      Exchange the information between $Q[i]$ and $R[(i + 2^{t-1}) \mod m]$
  ```

- **Invariant:**
  - Let $\alpha[i]$ be the information of $Q[i]$ and $R[i]$ after their initial exchange.
  - After round $t$ know nodes $Q[i]$ and $R[(i + 2^{t-1}) \mod m]$: $\cup_{0 \leq j \leq 2^{t-1}} \alpha[(i + j) \mod m]$
  - The invariant is easy to be shown.
Gossip on Graphs with $2 \cdot m$ Nodes

Theorem:

For all $m \in \mathbb{N}$ we have: $r_2(K(2m)) = \lceil \log 2m \rceil$

Proof: Split the nodes in groups $Q[i]$ and $R[i]$ ($0 \leq i \leq m - 1$).

- algorithm:
  
  for all $i \in \{0, \ldots, m - 1\}$ do in parallel
  
  Exchange the information between $Q[i]$ and $R[i]$

  for $t = 1$ to $\lceil \log_2 m \rceil$ do
    
    for all $i \in \{0, \ldots, m - 1\}$ do in parallel
      
      Exchange the information between $Q[i]$ and $R[(i + 2^{t-1}) \mod m]$

- Invariant:

  - Let $\alpha[i]$ be the information of $Q[i]$ and $R[i]$ after their initial exchange.
  
  - After round $t$ know nodes $Q[i]$ and $R[(i + 2^{t-1}) \mod m]$:
    
    $\bigcup_{0 \leq j \leq 2^{t-1}} \alpha[(i + j) \mod m]$

- The invariant is easy to be shown.
Gossip on Graphs with $2 \cdot m + 1$ Nodes (a try)

We need an extra round.

A nice proof with this idea will become complicated.

We will try to put some structure into the proof.
We need an extra round.

A nice proof with this idea will become complicated.

We will try to put some structure into the proof.
Gossip on Graphs with $2 \cdot m + 1$ Nodes (a try)

- We need an extra round.
- A nice proof with this idea will become complicated.
- We will try to put some structure into the proof.
Gossip on Graphs with $2 \cdot m + 1$ Nodes (a try)

- We need an extra round.
- A nice proof with this idea will become complicated.
- We will try to put some structure into the proof.
Gossip on Graphs with $2 \cdot m + 1$ Nodes (a try)

- We need an extra round.
- A nice proof with this idea will become complicated.
- We will try to put some structure into the proof.
Gossip on Graphs with $2 \cdot m + 1$ Nodes (a try)

- We need an extra round.
- A nice proof with this idea will become complicated.
- We will try to put some structure into the proof.
Gossip on Graphs with $2 \cdot m + 1$ Nodes (a try)

- We need an extra round.
- A nice proof with this idea will become complicated.
- We will try to put some structure into the proof.
Gossip on Graphs with $2 \cdot m + 1$ Nodes (a try)

- We need an extra round.
- A nice proof with this idea will become complicated.
- We will try to put some structure into the proof.
Gossip on Graphs with $2 \cdot m + 1$ Nodes (a try)

We need an extra round.

A nice proof with this idea will become complicated.

We will try to put some structure into the proof.
Gossip on Graphs with $2 \cdot m + 1$ Nodes (a try)

- We need an extra round.
- A nice proof with this idea will become complicated.
- We will try to put some structure into the proof.
We need an extra round.

A nice proof with this idea will become complicated.

We will try to put some structure into the proof.
Gossip on Graphs with $2 \cdot m + 1$ Nodes (a try)

- We need an extra round.
- A nice proof with this idea will become complicated.
- We will try to put some structure into the proof.
Gossip on Graphs with $2 \cdot m + 1$ Nodes (a try)

- We need an extra round.
- A nice proof with this idea will become complicated.
- We will try to put some structure into the proof.
Gossip on Graphs with $2 \cdot m + 1$ Nodes (Idea)

How could this be an idea?

- We only have the edges of the first step.
- Idea: We could now choose a small even number of Nodes, which together have the total information.
- These nodes may perform the above gossip algorithm.
- In the last step we repeat the first round.
Gossip on Graphs with $2 \cdot m + 1$ Nodes (Idea)

- How could this be an idea?
- We only have the edges of the first step.
- Idea: We could now choose a small even number of Nodes, which together have the total information.
- These nodes may perform the above gossip algorithm.
- In the last step we repeat the first round.
Gossip on Graphs with $2 \cdot m + 1$ Nodes (Idea)

How could this be an idea?

We only have the edges of the first step.

Idea: We could now choose a small even number of Nodes, which together have the total information.

These nodes may perform the above gossip algorithm.

In the last step we repeat the first round.
Gossip on Graphs with $2 \cdot m + 1$ Nodes (Idea)

- How could this be an idea?
- We only have the edges of the first step.
- Idea: We could now choose a small even number of Nodes, which together have the total information.
- These nodes may perform the above gossip algorithm.
- In the last step we repeat the first round.
Gossip on Graphs with $2 \cdot m + 1$ Nodes (Idea)

- How could this be an idea?
- We only have the edges of the first step.
- Idea: We could now choose a small even number of Nodes, which together have the total information.
- These nodes may perform the above gossip algorithm.
- In the last step we repeat the first round.
Let $n = 2 \cdot m + 1$.

Let $v_0, v_1, v_2, \ldots, v_{n-1}$ be all nodes.

For all $i \in \{0, 1, \ldots, m - 1\}$ the node $v_{m+2+i}$ sends to $v_i$.

The node $\{v_0, v_1, v_2, \ldots, v_m\}$ have now the total information.

If $m + 1$ is even, perform a gossip on the nodes $\{v_0, v_1, v_2, \ldots, v_m\}$.

If $m + 1$ is odd, perform a gossip on the nodes $\{v_0, v_1, v_2, \ldots, v_{m+1}\}$.

For all $i \in \{0, 1, \ldots, m - 1\}$ the nodes $v_i$ send to $v_{m+2+i}$.

Correctness follows direct by the construction.

Running time for $m + 1$ even:

$$r_2(K(m+1)) + 2 = \lceil \log_2 (m+1) \rceil + 2 = \lceil \log_2 \left( \frac{n+1}{2} \right) \rceil + 2$$

$$= \lceil \log_2 (n+1) \rceil + 1 = \lceil \log_2 n \rceil + 1$$

Running time for $m + 1$ odd:

$$r_2(K(m+2)) + 2 = \lceil \log_2 (m+2) \rceil + 2 = \lceil \log_2 \left( \frac{n+3}{2} \right) \rceil + 2$$

$$= \lceil \log_2 (n+3) \rceil + 1 = \lceil \log_2 n \rceil + 1$$
Gossip on Graphs with $2 \cdot m + 1$ Nodes

- Let $n = 2 \cdot m + 1$.
- Let $v_0, v_1, v_2, \cdots, v_{n-1}$ be all nodes.
- For all $i \in \{0, 1, \cdots, m - 1\}$ the node $v_{m+2+i}$ sends to $v_i$.
- The node $\{v_0, v_1, v_2, \cdots, v_m\}$ have now the total information.
- If $m + 1$ is even, perform a gossip on the nodes $\{v_0, v_1, v_2, \cdots, v_m\}$.
- If $m + 1$ is odd, perform a gossip on the nodes $\{v_0, v_1, v_2, \cdots, v_{m+1}\}$.
- For all $i \in \{0, 1, \cdots, m - 1\}$ the nodes $v_i$ send to $v_{m+2+i}$.
- Correctness follows direct by the construction.

Running time for $m + 1$ even:
\[
r_2(K(m + 1)) + 2 = \lceil \log_2(m + 1) \rceil + 2 = \lceil \log_2 \left( \frac{n+1}{2} \right) \rceil + 2
\]
\[
= \lceil \log_2(n + 1) \rceil + 1 = \lceil \log_2 n \rceil + 1
\]

Running time for $m + 1$ odd:
\[
r_2(K(m + 2)) + 2 = \lceil \log_2(m + 2) \rceil + 2 = \lceil \log_2 \left( \frac{n+3}{2} \right) \rceil + 2
\]
\[
= \lceil \log_2(n + 3) \rceil + 1 = \lceil \log_2 n \rceil + 1
\]
Gossip on Graphs with $2 \cdot m + 1$ Nodes

- Let $n = 2 \cdot m + 1$.
- Let $v_0, v_1, v_2, \cdots, v_{n-1}$ be all nodes.
- For all $i \in \{0, 1, \cdots, m - 1\}$ the node $v_{m+2+i}$ sends to $v_i$.
- The node $\{v_0, v_1, v_2, \cdots, v_m\}$ have now the total information.
- If $m + 1$ is even, perform a gossip on the nodes $\{v_0, v_1, v_2, \cdots, v_m\}$.
- If $m + 1$ is odd, perform a gossip on the nodes $\{v_0, v_1, v_2, \cdots, v_{m+1}\}$.
- For all $i \in \{0, 1, \cdots, m - 1\}$ the nodes $v_i$ send to $v_{m+2+i}$.
- Correctness follows direct by the construction.

Running time for $m + 1$ even:
$$r_2(K(m+1)) + 2 = [\log_2(m+1)] + 2 = [\log_2 (\frac{n+1}{2})] + 2 = [\log_2 n] + 1$$

Running time for $m + 1$ odd:
$$r_2(K(m+2)) + 2 = [\log_2(m+2)] + 2 = [\log_2 (\frac{n+3}{2})] + 2 = [\log_2 n] + 1$$
Gossip on Graphs with $2 \cdot m + 1$ Nodes

- Let $n = 2 \cdot m + 1$.
- Let $v_0, v_1, v_2, \cdots, v_{n-1}$ be all nodes.
- For all $i \in \{0, 1, \cdots, m - 1\}$ the node $v_{m+2+i}$ sends to $v_i$.
- The node $\{v_0, v_1, v_2, \cdots, v_m\}$ have now the total information.
- If $m + 1$ is even, perform a gossip on the nodes $\{v_0, v_1, v_2, \cdots, v_m\}$.
- If $m + 1$ is odd, perform a gossip on the nodes $\{v_0, v_1, v_2, \cdots, v_{m+1}\}$.
- For all $i \in \{0, 1, \cdots, m - 1\}$ the nodes $v_i$ send to $v_{m+2+i}$.
- Correctness follows direct by the construction.

Running time for $m + 1$ even:
\[
 r_2(K(m + 1)) + 2 = \lceil \log_2(m + 1) \rceil + 2 = \lceil \log_2 \left( \frac{n+1}{2} \right) \rceil + 2
\]
\[
 = \lceil \log_2(n + 1) \rceil + 1 = \lceil \log_2 n \rceil + 1
\]

Running time for $m + 1$ odd:
\[
 r_2(K(m + 2)) + 2 = \lceil \log_2(m + 2) \rceil + 2 = \lceil \log_2 \left( \frac{n+3}{2} \right) \rceil + 2
\]
\[
 = \lceil \log_2(n + 3) \rceil + 1 = \lceil \log_2 n \rceil + 1
\]
Let \( n = 2 \cdot m + 1 \).

Let \( v_0, v_1, v_2, \cdots, v_{n-1} \) be all nodes.

For all \( i \in \{0, 1, \cdots, m-1\} \) the node \( v_{m+2+i} \) sends to \( v_i \).

The node \( \{v_0, v_1, v_2, \cdots, v_m\} \) have now the total information.

If \( m + 1 \) is even, perform a gossip on the nodes \( \{v_0, v_1, v_2, \cdots, v_m\} \).

If \( m + 1 \) is odd, perform a gossip on the nodes \( \{v_0, v_1, v_2, \cdots, v_{m+1}\} \).

For all \( i \in \{0, 1, \cdots, m-1\} \) the nodes \( v_i \) send to \( v_{m+2+i} \).

Correctness follows direct by the construction.

Running time for \( m + 1 \) even:
\[
r_2(K(m+1)) + 2 = \left\lfloor \log_2(m+1) \right\rfloor + 2 = \left\lfloor \log_2 \left( \frac{n+1}{2} \right) \right\rfloor + 2
\]
\[
= \left\lfloor \log_2(n+1) \right\rfloor + 1 = \left\lfloor \log_2 n \right\rfloor + 1
\]

Running time for \( m + 1 \) odd:
\[
r_2(K(m+2)) + 2 = \left\lfloor \log_2(m+2) \right\rfloor + 2 = \left\lfloor \log_2 \left( \frac{n+3}{2} \right) \right\rfloor + 2
\]
\[
= \left\lfloor \log_2(n+3) \right\rfloor + 1 = \left\lfloor \log_2 n \right\rfloor + 1
\]
Gossip on Graphs with $2 \cdot m + 1$ Nodes

- Let $n = 2 \cdot m + 1$.
- Let $v_0, v_1, v_2, \cdots, v_{n-1}$ be all nodes.
- For all $i \in \{0, 1, \cdots, m - 1\}$ the node $v_{m+2+i}$ sends to $v_i$.
- The node $\{v_0, v_1, v_2, \cdots, v_m\}$ have now the total information.
- If $m + 1$ is even, perform a gossip on the nodes $\{v_0, v_1, v_2, \cdots, v_m\}$.
- If $m + 1$ is odd, perform a gossip on the nodes $\{v_0, v_1, v_2, \cdots, v_{m+1}\}$.
- For all $i \in \{0, 1, \cdots, m - 1\}$ the nodes $v_i$ send to $v_{m+2+i}$.
- Correctness follows direct by the construction.

Running time for $m + 1$ even:

\[
2(K(m+1)) + 2 = \left\lceil \log_2(m+1) \right\rceil + 2 = \left\lceil \log_2 \left(\frac{n+1}{2}\right) \right\rceil + 2 = \left\lceil \log_2 n \right\rceil + 1
\]

Running time for $m + 1$ odd:

\[
2(K(m+2)) + 2 = \left\lceil \log_2(m+2) \right\rceil + 2 = \left\lceil \log_2 \left(\frac{n+3}{2}\right) \right\rceil + 2 = \left\lceil \log_2 n \right\rceil + 1
\]
Gossip on Graphs with \( 2 \cdot m + 1 \) Nodes

- Let \( n = 2 \cdot m + 1 \).
- Let \( v_0, v_1, v_2, \ldots, v_{n-1} \) be all nodes.
- For all \( i \in \{0, 1, \ldots, m-1\} \) the node \( v_{m+2+i} \) sends to \( v_i \).
- The node \( \{v_0, v_1, v_2, \ldots, v_m\} \) have now the total information.
- If \( m + 1 \) is even, perform a gossip on the nodes \( \{v_0, v_1, v_2, \ldots, v_m\} \).
- If \( m + 1 \) is odd, perform a gossip on the nodes \( \{v_0, v_1, v_2, \ldots, v_{m+1}\} \).
- For all \( i \in \{0, 1, \ldots, m-1\} \) the nodes \( v_i \) send to \( v_{m+2+i} \).
- Correctness follows direct by the construction.

Running time for \( m + 1 \) even:
\[
r_2(K(m + 1)) + 2 = \lceil \log_2(m + 1) \rceil + 2 = \lceil \log_2 \left( \frac{n+1}{2} \right) \rceil + 2
\]

Running time for \( m + 1 \) odd:
\[
r_2(K(m + 2)) + 2 = \lceil \log_2(m + 2) \rceil + 2 = \lceil \log_2 \left( \frac{n+3}{2} \right) \rceil + 2
\]
Gossip on Graphs with $2 \cdot m + 1$ Nodes

- Let $n = 2 \cdot m + 1$.
- Let $v_0, v_1, v_2, \cdots, v_{n-1}$ be all nodes.
- For all $i \in \{0, 1, \cdots, m - 1\}$ the node $v_{m+2+i}$ sends to $v_i$.
- The node $\{v_0, v_1, v_2, \cdots, v_m\}$ have now the total information.
- If $m + 1$ is even, perform a gossip on the nodes $\{v_0, v_1, v_2, \cdots, v_m\}$.
- If $m + 1$ is odd, perform a gossip on the nodes $\{v_0, v_1, v_2, \cdots, v_{m+1}\}$.
- For all $i \in \{0, 1, \cdots, m - 1\}$ the nodes $v_i$ send to $v_{m+2+i}$.
- Correctness follows direct by the construction.

Running time for $m + 1$ even:

$$r_2(K(m + 1)) + 2 = \lceil \log_2(m + 1) \rceil + 2 = \lceil \log_2 (\frac{n+1}{2}) \rceil + 2$$

$$= \lceil \log_2(n + 1) \rceil + 1 = \lceil \log_2 n \rceil + 1$$

Running time for $m + 1$ odd:

$$r_2(K(m + 2)) + 2 = \lceil \log_2(m + 2) \rceil + 2 = \lceil \log_2 (\frac{n+3}{2}) \rceil + 2$$

$$= \lceil \log_2(n + 3) \rceil + 1 = \lceil \log_2 n \rceil + 1$$
Gossip on Graphs with $2 \cdot m + 1$ Nodes

- Let $n = 2 \cdot m + 1$.
- Let $v_0, v_1, v_2, \cdots, v_{n-1}$ be all nodes.
- For all $i \in \{0, 1, \cdots, m-1\}$ the node $v_{m+2+i}$ sends to $v_i$.
- The node $\{v_0, v_1, v_2, \cdots, v_m\}$ have now the total information.
- If $m + 1$ is even, perform a gossip on the nodes $\{v_0, v_1, v_2, \cdots, v_m\}$.
- If $m + 1$ is odd, perform a gossip on the nodes $\{v_0, v_1, v_2, \cdots, v_{m+1}\}$.
- For all $i \in \{0, 1, \cdots, m-1\}$ the nodes $v_i$ send to $v_{m+2+i}$.
- Correctness follows direct by the construction.

Running time for $m + 1$ even:

$$r_2(K(m + 1)) + 2 = \lceil \log_2(m + 1) \rceil + 2 = \lceil \log_2 \left( \frac{n+1}{2} \right) \rceil + 2$$

Running time for $m + 1$ odd:

$$r_2(K(m + 2)) + 2 = \lceil \log_2(m + 2) \rceil + 2 = \lceil \log_2 \left( \frac{n+3}{2} \right) \rceil + 2$$
Gossip on Graphs with $2 \cdot m + 1$ Nodes

- Let $n = 2 \cdot m + 1$.
- Let $v_0, v_1, v_2, \cdots, v_{n-1}$ be all nodes.
- For all $i \in \{0, 1, \cdots, m - 1\}$ the node $v_{m+2+i}$ sends to $v_i$.
- The node $\{v_0, v_1, v_2, \cdots, v_m\}$ have now the total information.
- If $m + 1$ is even, perform a gossip on the nodes $\{v_0, v_1, v_2, \cdots, v_m\}$.
- If $m + 1$ is odd, perform a gossip on the nodes $\{v_0, v_1, v_2, \cdots, v_{m+1}\}$.
- For all $i \in \{0, 1, \cdots, m - 1\}$ the nodes $v_i$ send to $v_{m+2+i}$.
- Correctness follows direct by the construction.

Running time for $m + 1$ even:
$$r_2(K(m + 1)) + 2 = \lfloor \log_2(m + 1) \rfloor + 2 = \lfloor \log_2 (\frac{n+1}{2}) \rfloor + 2$$
$$= \lfloor \log_2(n + 1) \rfloor + 1 = \lfloor \log_2 n \rfloor + 1$$

Running time for $m + 1$ odd:
$$r_2(K(m + 2)) + 2 = \lfloor \log_2(m + 2) \rfloor + 2 = \lfloor \log_2 (\frac{n+3}{2}) \rfloor + 2$$
$$= \lfloor \log_2(n + 3) \rfloor + 1 = \lfloor \log_2 n \rfloor + 1$$
1st Idea (Let the Knowledge grow)

- We need more rounds.
- A nice proof with this idea will become complicated.
- We will try to put some structure into the proof.
We need more rounds.

A nice proof with this idea will become complicated.

We will try to put some structure into the proof.
1\textsuperscript{st} Idea (Let the Knowledge grow)

- We need more rounds.
- A nice proof with this idea will become complicated.
- We will try to put some structure into the proof.
1st Idea (Let the Knowledge grow)

- We need more rounds.
- A nice proof with this idea will become complicated.
- We will try to put some structure into the proof.
1st Idea (Let the Knowledge grow)

- We need more rounds.
- A nice proof with this idea will become complicated.
- We will try to put some structure into the proof.
1st Idea (Let the Knowledge grow)

- We need more rounds.
- A nice proof with this idea will become complicated.
- We will try to put some structure into the proof.
We need more rounds.

A nice proof with this idea will become complicated.

We will try to put some structure into the proof.
1\textsuperscript{st} Idea (Let the Knowledge grow)

- We need more rounds.
- A nice proof with this idea will become complicated.
- We will try to put some structure into the proof.
1st Idea (Let the Knowledge grow)

- We need more rounds.
- A nice proof with this idea will become complicated.
- We will try to put some structure into the proof.
1\textsuperscript{st} Idea (Let the Knowledge grow)

- We need more rounds.
- A nice proof with this idea will become complicated.
- We will try to put some structure into the proof.
1st Idea (Let the Knowledge grow)

We need more rounds.
A nice proof with this idea will become complicated.
We will try to put some structure into the proof.
1st Idea (Let the Knowledge grow)

- We need more rounds.
- A nice proof with this idea will become complicated.
- We will try to put some structure into the proof.
1st Idea (Let the Knowledge grow)

- We need more rounds.
- A nice proof with this idea will become complicated.
- We will try to put some structure into the proof.
We need an additional two rounds.

\( v_x \) and \( w_y \) alternate as sender and receiver.

The information grows in blocks (intervals) in the nodes.

With this idea we may do the proof.

Only the first two rounds are special.
2\textsuperscript{nd} Idea (Let the Knowledge grow in a structured way)

- We need an additional two rounds.
- $v_x$ and $w_y$ alternate as sender and receiver.
- The information grows in blocks (intervals) in the nodes.
- With this idea we may do the proof.
- Only the first two rounds are special.
2nd Idea (Let the Knowledge grow in a structured way)

- We need an additional two rounds.
- $v_x$ and $w_y$ alternate as sender and receiver.
- The information grows in blocks (intervals) in the nodes.
- With this idea we may do the proof.
- Only the first two rounds are special.
We need an additional two rounds.

\( v_x \) and \( w_y \) alternate as sender and receiver.

The information grows in blocks (intervals) in the nodes.

With this idea we may do the proof.

Only the first two rounds are special.
2nd Idea (Let the Knowledge grow in a structured way)

- We need an additional two rounds.
- $v_x$ and $w_y$ alternate as sender and receiver.
- The information grows in blocks (intervals) in the nodes.
- With this idea we may do the proof.
- Only the first two rounds are special.
2\textsuperscript{nd} Idea (Let the Knowledge grow in a structured way)

- We need an additional two rounds.
- $v_x$ and $w_y$ alternate as sender and receiver.
- The information grows in blocks (intervals) in the nodes.
- With this idea we may do the proof.
- Only the first two rounds are special.
2\textsuperscript{nd} Idea (Let the Knowledge grow in a structured way)

- We need an additional two rounds.
- $v_x$ and $w_y$ alternate as sender and receiver.
- The information grows in blocks (intervals) in the nodes.
- With this idea we may do the proof.
- Only the first two rounds are special.
We need an additional two rounds.

\( v_x \) and \( w_y \) alternate as sender and receiver.

The information grows in blocks (intervals) in the nodes.

With this idea we may do the proof.

Only the first two rounds are special.
2\textsuperscript{nd} Idea (Let the Knowledge grow in a structured way)

- We need an additional two rounds.
- $v_x$ and $w_y$ alternate as sender and receiver.
- The information grows in blocks (intervals) in the nodes.
- With this idea we may do the proof.
- Only the first two rounds are special.
We need an additional two rounds.

$\nu_x$ and $w_y$ alternate as sender and receiver.

The information grows in blocks (intervals) in the nodes.

With this idea we may do the proof.

Only the first two rounds are special.
We need an additional two rounds.

\(v_x\) and \(w_y\) alternate as sender and receiver.

The information grows in blocks (intervals) in the nodes.

With this idea we may do the proof.

Only the first two rounds are special.
We need an additional two rounds.

\(v_x\) and \(w_y\) alternate as sender and receiver.

The information grows in blocks (intervals) in the nodes.

With this idea we may do the proof.

Only the first two rounds are special.
2\textsuperscript{nd} Idea (Let the Knowledge grow in a structured way)

- We need an additional two rounds.
- \(v_x\) and \(w_y\) alternate as sender and receiver.
- The information grows in blocks (intervals) in the nodes.
- With this idea we may do the proof.
- Only the first two rounds are special.
We need an additional two rounds.
\( v_x \) and \( w_y \) alternate as sender and receiver.
The information grows in blocks (intervals) in the nodes.
With this idea we may do the proof.
Only the first two rounds are special.
We need an additional two rounds.
- \( v_x \) and \( w_y \) alternate as sender and receiver.
- The information grows in blocks (intervals) in the nodes.
- With this idea we may do the proof.
- Only the first two rounds are special.
2\textsuperscript{nd} Idea (Let the Knowledge grow in a structured way)

- We need an additional two rounds.
- $v_x$ and $w_y$ alternate as sender and receiver.
- The information grows in blocks (intervals) in the nodes.
- With this idea we may do the proof.
- Only the first two rounds are special.
2nd Idea (Let the Knowledge grow in a structured way)

- After the first two rounds some node-pairs share their information.
- Consider this situation as the start:
  - All $v_x$ and $w_x$ have one information pair.
  - $v_i$ sends to $w_j$ and the $w_x$ have 2 information pairs.
  - $w_i$ sends to $v_j$ and the $v_x$ have 3 information pairs.
  - $v_i$ sends to $w_j$ and the $w_x$ have 5 information pairs.
  - $w_i$ sends to $v_j$ and the $v_x$ have 8 information pairs.
  - $v_i$ sends to $w_j$ and the $w_x$ have 13 information pairs.
  - $w_i$ sends to $v_j$ and the $v_x$ have 21 information pairs.
  - Thus the grow-rate and the algorithm is clearly visible.
2nd Idea (Let the Knowledge grow in a structured way)

- After the first two rounds some node-pairs share their information.

- Consider this situation as the start:
  - All $v_x$ and $w_x$ have one information pair.
  - $v_i$ sends to $w_j$ and the $w_x$ have 2 information pairs.
  - $w_i$ sends to $v_j$ and the $v_x$ have 3 information pairs.
  - $v_i$ sends to $w_j$ and the $w_x$ have 5 information pairs.
  - $w_i$ sends to $v_j$ and the $v_x$ have 8 information pairs.
  - $v_i$ sends to $w_j$ and the $w_x$ have 13 information pairs.
  - $w_i$ sends to $v_j$ and the $v_x$ have 21 information pairs.
  - Thus the grow-rate and the algorithm is clearly visible.
2\textsuperscript{nd} Idea (Let the Knowledge grow in a structured way)

- After the first two rounds some node-pairs share their information.
- Consider this situation as the start:
  - All $v_x$ and $w_x$ have one information pair.
  - $v_i$ sends to $w_j$ and the $w_x$ have 2 information pairs.
  - $w_i$ sends to $v_j$ and the $v_x$ have 3 information pairs.
  - $v_i$ sends to $w_j$ and the $w_x$ have 5 information pairs.
  - $w_i$ sends to $v_j$ and the $v_x$ have 8 information pairs.
  - $v_i$ sends to $w_j$ and the $w_x$ have 13 information pairs.
  - $w_i$ sends to $v_j$ and the $v_x$ have 21 information pairs.
- Thus the grow-rate and the algorithm is clearly visible.
2nd Idea (Let the Knowledge grow in a structured way)

- After the first two rounds some node-pairs share their information.
- Consider this situation as the start:
  - All $v_x$ and $w_x$ have one information pair.
  - $v_i$ sends to $w_j$ and the $w_x$ have 2 information pairs.
  - $w_i$ sends to $v_j$ and the $v_x$ have 3 information pairs.
  - $v_i$ sends to $w_j$ and the $w_x$ have 5 information pairs.
  - $w_i$ sends to $v_j$ and the $v_x$ have 8 information pairs.
  - $v_i$ sends to $w_j$ and the $w_x$ have 13 information pairs.
  - $w_i$ sends to $v_j$ and the $v_x$ have 21 information pairs.
- Thus the grow-rate and the algorithm is clearly visible.
2nd Idea (Let the Knowledge grow in a structured way)

- After the first two rounds some node-pairs share their information.
- Consider this situation as the start:
  - All $v_x$ and $w_x$ have one information pair.
  - $v_i$ sends to $w_j$ and the $w_x$ have 2 information pairs.
  - $w_i$ sends to $v_j$ and the $v_x$ have 3 information pairs.
  - $v_i$ sends to $w_j$ and the $w_x$ have 5 information pairs.
  - $w_i$ sends to $v_j$ and the $v_x$ have 8 information pairs.
  - $v_i$ sends to $w_j$ and the $w_x$ have 13 information pairs.
  - $w_i$ sends to $v_j$ and the $v_x$ have 21 information pairs.
  - Thus the grow-rate and the algorithm is clearly visible.
2nd Idea (Let the Knowledge grow in a structured way)

- After the first two rounds some node-pairs share their information.
- Consider this situation as the start:
  - All $v_x$ and $w_x$ have one information pair.
  - $v_i$ sends to $w_j$ and the $w_x$ have 2 information pairs.
  - $w_i$ sends to $v_j$ and the $v_x$ have 3 information pairs.
  - $v_i$ sends to $w_j$ and the $w_x$ have 5 information pairs.
  - $w_i$ sends to $v_j$ and the $v_x$ have 8 information pairs.
  - $v_i$ sends to $w_j$ and the $w_x$ have 13 information pairs.
  - $w_i$ sends to $v_j$ and the $v_x$ have 21 information pairs.
- Thus the grow-rate and the algorithm is clearly visible.
2\textsuperscript{nd} Idea (Let the Knowledge grow in a structured way)

- After the first two rounds some node-pairs share their information.
- Consider this situation as the start:
  - All $v_x$ and $w_x$ have one information pair.
  - $v_i$ sends to $w_j$ and the $w_x$ have 2 information pairs.
  - $w_i$ sends to $v_j$ and the $v_x$ have 3 information pairs.
  - $v_i$ sends to $w_j$ and the $w_x$ have 5 information pairs.
  - $w_i$ sends to $v_j$ and the $v_x$ have 8 information pairs.
  - $v_i$ sends to $w_j$ and the $w_x$ have 13 information pairs.
  - $w_i$ sends to $v_j$ and the $v_x$ have 21 information pairs.
  - Thus the grow-rate and the algorithm is clearly visible.
2\textsuperscript{nd} Idea (Let the Knowledge grow in a structured way)

- After the first two rounds some node-pairs share their information.
- Consider this situation as the start:
  - All $v_x$ and $w_x$ have one information pair.
  - $v_i$ sends to $w_j$ and the $w_x$ have 2 information pairs.
  - $w_i$ sends to $v_j$ and the $v_x$ have 3 information pairs.
  - $v_i$ sends to $w_j$ and the $w_x$ have 5 information pairs.
  - $w_i$ sends to $v_j$ and the $v_x$ have 8 information pairs.
  - $v_i$ sends to $w_j$ and the $w_x$ have 13 information pairs.
  - $w_i$ sends to $v_j$ and the $v_x$ have 21 information pairs.
  - Thus the grow-rate and the algorithm is clearly visible.
2nd Idea (Let the Knowledge grow in a structured way)

- After the first two rounds some node-pairs share their information.
- Consider this situation as the start:
  - All \( v_x \) and \( w_x \) have one information pair.
  - \( v_i \) sends to \( w_j \) and the \( w_x \) have 2 information pairs.
  - \( w_i \) sends to \( v_j \) and the \( v_x \) have 3 information pairs.
  - \( v_i \) sends to \( w_j \) and the \( w_x \) have 5 information pairs.
  - \( w_i \) sends to \( v_j \) and the \( v_x \) have 8 information pairs.
  - \( v_i \) sends to \( w_j \) and the \( w_x \) have 13 information pairs.
  - \( w_i \) sends to \( v_j \) and the \( v_x \) have 21 information pairs.
- Thus the grow-rate and the algorithm is clearly visible.
After the first two rounds some node-pairs share their information.

Consider this situation as the start:

- All \( v_x \) and \( w_x \) have one information pair.
- \( v_i \) sends to \( w_j \) and the \( w_x \) have 2 information pairs.
- \( w_i \) sends to \( v_j \) and the \( v_x \) have 3 information pairs.
- \( v_i \) sends to \( w_j \) and the \( w_x \) have 5 information pairs.
- \( w_i \) sends to \( v_j \) and the \( v_x \) have 8 information pairs.
- \( v_i \) sends to \( w_j \) and the \( w_x \) have 13 information pairs.
- \( w_i \) sends to \( v_j \) and the \( v_x \) have 21 information pairs.
- Thus the grow-rate and the algorithm is clearly visible.
algorithm

Let $n = 2m$.

Gossip-Algorithm:

1. $t := 0$;
2. for all $i \in \{0, \ldots, m - 1\}$ do in parallel $R[i]$ sends to $Q[i]$;
3. for all $i \in \{0, \ldots, m - 1\}$ do in parallel $Q[i]$ sends to $R[i]$;
4. while $fib(2t + 1) < m$ do begin
   1. $t := t + 1$;
   2. for all $i \in \{0, \ldots, m - 1\}$ do in parallel $R[(i + fib(2t - 1)) \mod m]$ sends to $Q[i]$;
   3. if $fib(2t) < m$ then
      1. for all $i \in \{0, \ldots, m - 1\}$ do in parallel $Q[(i + fib(2t)) \mod m]$ sends to $R[i]$
end;
Let $n = 2m$.

Gossip-Algorithm:

- $t := 0$;
- for all $i \in \{0, \ldots, m - 1\}$ do in parallel $R[i]$ sends to $Q[i]$;
- for all $i \in \{0, \ldots, m - 1\}$ do in parallel $Q[i]$ sends to $R[i]$;
- while $\text{fib}(2t + 1) < m$ do begin
  - $t := t + 1$;
  - for all $i \in \{0, \ldots, m - 1\}$ do in parallel
    - $R[(i + \text{fib}(2t - 1)) \mod m]$ sends to $Q[i]$;
  - if $\text{fib}(2t) < m$ then
    - for all $i \in \{0, \ldots, m - 1\}$ do in parallel
      - $Q[(i + \text{fib}(2t)) \mod m]$ sends to $R[i]$;
  - end;

\[
\begin{align*}
\text{fib}(0) &= \text{fib}(1) = 1 \\
\text{fib}(i) &= \text{fib}(i-1) + \text{fib}(i-2)
\end{align*}
\]
algorithm

- Let $n = 2m$.
- Gossip-Algorithm:
  
  $t := 0$
  
  for all $i \in \{0, \ldots, m - 1\}$ do in parallel $R[i]$ sends to $Q[i]$;
  
  for all $i \in \{0, \ldots, m - 1\}$ do in parallel $Q[i]$ sends to $R[i]$;
  
  while $fib(2t + 1) < m$ do begin
    $t := t + 1$
    
    for all $i \in \{0, \ldots, m - 1\}$ do in parallel
    
    $R[(i + fib(2t - 1)) \mod m]$ sends to $Q[i]$;
    
    if $fib(2t) < m$ then
      
      for all $i \in \{0, \ldots, m - 1\}$ do in parallel
      
      $Q[(i + fib(2t)) \mod m]$ sends to $R[i]$;
  
  end;

\[
\begin{align*}
    fib(0) &= fib(1) = 1 \\
    fib(i) &= fib(i - 1) + fib(i - 2)
\end{align*}
\]
Let $n = 2m$.

**Gossip-Algorithm:**

- $t := 0$;
- for all $i \in \{0, \ldots, m-1\}$ do in parallel $R[i]$ sends to $Q[i]$;
- for all $i \in \{0, \ldots, m-1\}$ do in parallel $Q[i]$ sends to $R[i]$;
- while $fib(2t + 1) < m$ do begin
  - $t := t + 1$;
  - for all $i \in \{0, \ldots, m-1\}$ do in parallel $R[(i + fib(2t-1)) \mod m]$ sends to $Q[i]$;
  - if $fib(2t) < m$ then
    - for all $i \in \{0, \ldots, m-1\}$ do in parallel $Q[(i + fib(2t)) \mod m]$ sends to $R[i]$  

end;
algorithm

- Let $n = 2m$.

- Gossip-Algorithm:
  
  $t := 0$;
  
  for all $i \in \{0, \ldots, m - 1\}$ do in parallel $R[i]$ sends to $Q[i]$;
  
  for all $i \in \{0, \ldots, m - 1\}$ do in parallel $Q[i]$ sends to $R[i]$;
  
  while $\text{fib}(2t + 1) < m$ do begin
    
    $t := t + 1$;
    
    for all $i \in \{0, \ldots, m - 1\}$ do in parallel
    
    $R[(i + \text{fib}(2t - 1)) \mod m]$ sends to $Q[i]$;
    
    if $\text{fib}(2t) < m$ then
    
    for all $i \in \{0, \ldots, m - 1\}$ do in parallel
    
    $Q[(i + \text{fib}(2t)) \mod m]$ sends to $R[i]$

end;

$\text{fib}(0) = \text{fib}(1) = 1$

$\text{fib}(i) = \text{fib}(i - 1) + \text{fib}(i - 2)$
One-Way-Gossip

Theorem:
Let $n = 2m$ and $k = \min\{x \mid \text{fib}(x) \geq m\}$. Then we have $r(K(n)) \leq k + 1$.

Proof:
- The algorithm stops, if $\text{fib}(2t + 1) \geq m$ or $\text{fib}(2t) \geq m$ holds.
- The number of rounds within the loop is $2t$ or $2(t - 1) + 1$.
- The total number of rounds is $(k - 1) + 2$.
- Correctness may be proven by the following invariant:
- Let $a[i]$ be the information, which share $R[i]$ and $Q[i]$ after two rounds.
- After $t$ loops we have:
  - $Q[i]$ knows $\cup_{0 \leq j \leq \text{fib}(2t+1)-1} \alpha[(i + j) \mod m]$
  - $R[i]$ knows $\cup_{0 \leq j \leq \text{fib}(2t+2)-1} \alpha[(i + j) \mod m]$
- The correctness is a direct result of this.

\[ \begin{align*}
\text{fib}(0) &= 1 \\
\text{fib}(1) &= 1 \\
\text{fib}(i) &= \text{fib}(i - 1) + \text{fib}(i - 2)
\end{align*} \]
One-Way-Gossip

Theorem:
Let $n = 2m$ and $k = \min\{x \mid \text{fib}(x) \geq m\}$. Then we have $r(K(n)) \leq k + 1$.

Proof:
- The algorithm stops, if $\text{fib}(2t + 1) \geq m$ or $\text{fib}(2t) \geq m$ holds.
- The number of rounds within the loop is $2t$ or $2(t − 1) + 1$.
- The total number of rounds is $(k − 1) + 2$.
- Correctness may be proven by the following invariant:
- Let $a[i]$ be the information, which share $R[i]$ and $Q[i]$ after two rounds.
- After $t$ loops we have:
  - $Q[i]$ knows $\cup_{0 \leq j \leq \text{fib}(2t+1)−1} \alpha[(i + j) \mod m]$
  - $R[i]$ knows $\cup_{0 \leq j \leq \text{fib}(2t+2)−1} \alpha[(i + j) \mod m]$
- The correctness is a direct result of this.
One-Way-Gossip

Theorem:
Let $n = 2m$ and $k = \min\{x \mid fib(x) \geq m\}$. Then we have $r(K(n)) \leq k + 1$.

Proof:
- The algorithm stops, if $fib(2t + 1) \geq m$ or $fib(2t) \geq m$ holds.
- The number of rounds within the loop is $2t$ or $2(t - 1) + 1$.
- The total number of rounds is $(k - 1) + 2$.
- Correctness may be proven by the following invariant:
- Let $a[i]$ be the information, which share $R[i]$ and $Q[i]$ after two rounds.
- After $t$ loops we have:
  - $Q[i]$ knows $\bigcup_{0 \leq j \leq fib(2t+1)-1} \alpha[(i + j) \mod m]$
  - $R[i]$ knows $\bigcup_{0 \leq j \leq fib(2t+2)-1} \alpha[(i + j) \mod m]$
- The correctness is a direct result of this.
One-Way-Gossip

Theorem:
Let $n = 2m$ and $k = \min\{x \mid \text{fib}(x) \geq m\}$. Then we have $r(K(n)) \leq k + 1$.

Proof:
- The algorithm stops, if $\text{fib}(2t + 1) \geq m$ or $\text{fib}(2t) \geq m$ holds.
- The number of rounds within the loop is $2t$ or $2(t - 1) + 1$.
- The total number of rounds is $(k - 1) + 2$.
- Correctness may be proven by the following invariant:
  - Let $a[i]$ be the information, which share $R[i]$ and $Q[i]$ after two rounds.
  - After $t$ loops we have:
    - $Q[i]$ knows $\cup_{0 \leq j \leq \text{fib}(2t+1)-1} \alpha[(i + j) \mod m]$
    - $R[i]$ knows $\cup_{0 \leq j \leq \text{fib}(2t+2)-1} \alpha[(i + j) \mod m]$
- The correctness is a direct result of this.
One-Way-Gossip

Theorem:
Let $n = 2m$ and $k = \min\{x \mid \text{fib}(x) \geq m\}$. Then we have $r(K(n)) \leq k + 1$.

Proof:
- The algorithm stops, if $\text{fib}(2t + 1) \geq m$ or $\text{fib}(2t) \geq m$ holds.
- The number of rounds within the loop is $2t$ or $2(t - 1) + 1$.
- The total number of rounds is $(k - 1) + 2$.
- Correctness may be proven by the following invariant:
  - Let $a[i]$ be the information, which share $R[i]$ and $Q[i]$ after two rounds.
  - After $t$ loops we have:
    - $Q[i]$ knows $\bigcup_{0 \leq j \leq \text{fib}(2t + 1) - 1} \alpha[(i + j) \mod m]$
    - $R[i]$ knows $\bigcup_{0 \leq j \leq \text{fib}(2t + 2) - 1} \alpha[(i + j) \mod m]$
  - The correctness is a direct result of this.
Theorem:

Let \( n = 2m \) and \( k = \min\{x \mid \text{fib}(x) \geq m\} \). Then we have \( r(K(n)) \leq k + 1 \).

Proof:

- The algorithm stops, if \( \text{fib}(2t + 1) \geq m \) or \( \text{fib}(2t) \geq m \) holds.
- The number of rounds within the loop is \( 2t \) or \( 2(t - 1) + 1 \).
- The total number of rounds is \( (k - 1) + 2 \).
- Correctness may be proven by the following invariant:
  - Let \( a[i] \) be the information, which share \( R[i] \) and \( Q[i] \) after two rounds.
  - After \( t \) loops we have:
    - \( Q[i] \) knows \( \bigcup_{0 \leq j \leq \text{fib}(2t+1) - 1} \alpha[(i + j) \mod m] \)
    - \( R[i] \) knows \( \bigcup_{0 \leq j \leq \text{fib}(2t+2) - 1} \alpha[(i + j) \mod m] \)
  - The correctness is a direct result of this.
One-Way-Gossip

Theorem:
Let $n = 2m$ and $k = \min\{x \mid \text{fib}(x) \geq m\}$. Then we have $r(K(n)) \leq k + 1$.

Proof:

- The algorithm stops, if $\text{fib}(2t + 1) \geq m$ or $\text{fib}(2t) \geq m$ holds.
- The number of rounds within the loop is $2t$ or $2(t - 1) + 1$.
- The total number of rounds is $(k - 1) + 2$.
- Correctness may be proven by the following invariant:
  Let $a[i]$ be the information, which share $R[i]$ and $Q[i]$ after two rounds.
  After $t$ loops we have:
    - $Q[i]$ knows $\cup_{0 \leq j \leq \text{fib}(2t+1) - 1} \alpha[(i + j) \mod m]$
    - $R[i]$ knows $\cup_{0 \leq j \leq \text{fib}(2t+2) - 1} \alpha[(i + j) \mod m]$
- The correctness is a direct result of this.
One-Way-Gossip

Theorem:
Let $n = 2m$ and $k = \min\{x \mid \text{fib}(x) \geq m\}$. Then we have $r(K(n)) \leq k + 1$.

Proof:
- The algorithm stops, if $\text{fib}(2t + 1) \geq m$ or $\text{fib}(2t) \geq m$ holds.
- The number of rounds within the loop is $2t$ or $2(t - 1) + 1$.
- The total number of rounds is $(k - 1) + 2$.
- Correctness may be proven by the following invariant:
- Let $a[i]$ be the information, which share $R[i]$ and $Q[i]$ after two rounds.
- After $t$ loops we have:
  - $Q[i]$ knows $\cup_{0 \leq j \leq \text{fib}(2t+1)-1} \alpha[(i + j) \mod m]$
  - $R[i]$ knows $\cup_{0 \leq j \leq \text{fib}(2t+2)-1} \alpha[(i + j) \mod m]$
- The correctness is a direct result of this.
One-Way-Gossip

Theorem:

Let $n = 2m$ and $k = \min\{x \mid \text{fib}(x) \geq m\}$. Then we have $r(K(n)) \leq k + 1$.

Proof:

- The algorithm stops, if $\text{fib}(2t + 1) \geq m$ or $\text{fib}(2t) \geq m$ holds.
- The number of rounds within the loop is $2t$ or $2(t - 1) + 1$.
- The total number of rounds is $(k - 1) + 2$.
- Correctness may be proven by the following invariant:
- Let $a[i]$ be the information, which share $R[i]$ and $Q[i]$ after two rounds.
- After $t$ loops we have:
  - $Q[i]$ knows $\cup_{0 \leq j \leq \text{fib}(2t+1)-1} \alpha[(i + j) \mod m]$
  - $R[i]$ knows $\cup_{0 \leq j \leq \text{fib}(2t+2)-1} \alpha[(i + j) \mod m]$
- The correctness is a direct result of this.
One-Way-Gossip

Theorem:
Let \( n = 2m \) and \( k = \min\{x \mid \text{fib}(x) \geq m\} \). Then we have \( r(K(n)) \leq k + 1 \).

Proof:

- The algorithm stops, if \( \text{fib}(2t + 1) \geq m \) or \( \text{fib}(2t) \geq m \) holds.
- The number of rounds within the loop is \( 2t \) or \( 2(t - 1) + 1 \).
- The total number of rounds is \( (k - 1) + 2 \).
- Correctness may be proven by the following invariant:
- Let \( a[i] \) be the information, which share \( R[i] \) and \( Q[i] \) after two rounds.
- After \( t \) loops we have:
  - \( Q[i] \) knows \( \bigcup_{0 \leq j \leq \text{fib}(2t+1)-1} \alpha[(i + j) \mod m] \)
  - \( R[i] \) knows \( \bigcup_{0 \leq j \leq \text{fib}(2t+2)-1} \alpha[(i + j) \mod m] \)
- The correctness is a direct result of this.
One-Way-Gossip

**Theorem:**
Let $n = 2m - 1$ and $k = \min\{x \mid \text{fib}(x) \geq m\}$. Then we have $r(K(n)) \leq k + 2$.

**Proof:** Using the same idea as for the two-way mode.

**Theorem:**
Let $n$ even. Then we have: $r(K(n)) \geq 2 + \lceil \log_{\frac{1}{2}(1+\sqrt{5})} \frac{n}{2} \rceil$.

**Proof:** See literature (Idea is given the following).

<table>
<thead>
<tr>
<th>$n$</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
<th>14</th>
<th>16</th>
<th>18</th>
<th>20</th>
<th>22</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upper Bound</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Lower Bound</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>
One-Way-Gossip

Theorem:
Let \( n = 2m - 1 \) and \( k = \min\{x \mid \text{fib}(x) \geq m\} \). Then we have \( r(K(n)) \leq k + 2 \).

Proof: Using the same idea as for the two-way mode.

Theorem:
Let \( n \) even. Then we have: \( r(K(n)) \geq 2 + \lceil \log_{\frac{1}{2}}(1+\sqrt{5}) \frac{n}{2} \rceil \).

Proof: See literature (Idea is given the following).

<table>
<thead>
<tr>
<th>( n )</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
<th>14</th>
<th>16</th>
<th>18</th>
<th>20</th>
<th>22</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upper Bound</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Lower Bound</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>
One-Way-Gossip

Theorem:
Let $n = 2m - 1$ and $k = \min\{x \mid \text{fib}(x) \geq m\}$. Then we have $r(K(n)) \leq k + 2$.

Proof: Using the same idea as for the two-way mode.

Theorem:
Let $n$ even. Then we have: $r(K(n)) \geq 2 + \lceil \log_{\frac{1}{2}(1+\sqrt{5})} \frac{n}{2} \rceil$.

Proof: See literature (Idea is given the following).

<table>
<thead>
<tr>
<th>$n$</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
<th>14</th>
<th>16</th>
<th>18</th>
<th>20</th>
<th>22</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upper Bound</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Lower Bound</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>
Idea for the lower Bound

- Situation:
  - Algorithm with “fibonacci growth”.
  - No idea to enlarge this growth.

- Construction of a lower bound:
  - Start with an arbitrary algorithm.
  - Use only the restriction of the algorithm.
  - Abstract.

- We will now try to do the abstraction.

- Try the get the core-problem.

- The core-problem ist:
  - “Fibonacci growth” could not be improved.
Idea for the lower Bound

- **Situation:**
  - Algorithm with “fibonacci growth”.
  - No idea to enlarge this growth.

- **Construction of a lower bound:**
  - Start with an arbitrary algorithm.
  - Use only the restriction of the algorithm.
  - Abstract.

- We will now try to do the abstraction.

- Try the get the core-problem.

- The core-problem ist:
  - “Fibonacci growth” could not be improved.
Idea for the lower Bound

- **Situation:**
  - Algorithm with “fibonacci growth”.
  - No idea to enlarge this growth.

- **Construction of a lower bound:**
  - Start with an arbitrary algorithm.
  - Use only the restriction of the algorithm.
  - Abstract.

  We will now try to do the abstraction.

  Try the get the core-problem.

  The core-problem ist:
  - “Fibonacci growth” could not be improved.
Idea for the lower Bound

- **Situation:**
  - Algorithm with “fibonacci growth”.
  - No idea to enlarge this growth.

- **Construction of a lower bound:**
  - Start with an arbitrary algorithm.
  - Use only the restriction of the algorithm.
  - Abstract.

- We will now try to do the abstraction.
- Try the get the core-problem.
- The core-problem ist:
  - “Fibonacci growth” could not be improved.
Idea for the lower Bound

**Situation:**
- Algorithm with “fibonacci growth”.
- No idea to enlarge this growth.

**Construction of a lower bound:**
- **Start with an arbitrary algorithm.**
  - Use only the restriction of the algorithm.
  - Abstract.

We will now try to do the abstraction.

Try the get the core-problem.

The core-problem ist:
- “Fibonacci growth” could not be improved.
Idea for the lower Bound

- **Situation:**
  - Algorithm with “fibonacci growth”.
  - No idea to enlarge this growth.

- **Construction of a lower bound:**
  - Start with an arbitrary algorithm.
  - Use only the restriction of the algorithm.
  - Abstract.

- We will now try to do the abstraction.
- Try the get the core-problem.
- The core-problem ist:
  - “Fibonacci growth” could not be improved.
Idea for the lower Bound

- **Situation:**
  - Algorithm with “fibonacci growth”.
  - No idea to enlarge this growth.

- **Construction of a lower bound:**
  - Start with an arbitrary algorithm.
  - Use only the restriction of the algorithm.
  - Abstract.

- We will now try to do the abstraction.

- Try the get the core-problem.

- The core-problem ist:
  - “Fibonacci growth” could not be improved.
Idea for the lower Bound

- **Situation:**
  - Algorithm with “fibonacci growth”.
  - No idea to enlarge this growth.

- **Construction of a lower bound:**
  - Start with an arbitrary algorithm.
  - Use only the restriction of the algorithm.
  - Abstract.

- **We will now try to do the abstraction.**

- Try the get the core-problem.

- The core-problem ist:
  - “Fibonacci growth” could not be improved.
Idea for the lower Bound

- **Situation:**
  - Algorithm with “fibonacci growth”.
  - No idea to enlarge this growth.

- **Construction of a lower bound:**
  - Start with an arbitrary algorithm.
  - Use only the restriction of the algorithm.
  - Abstract.

- **We will now try to do the abstraction.**

- **Try the get the core-problem.**

- **The core-problem ist:**
  - “Fibonacci growth” could not be improved.
Idea for the lower Bound

Situation:
- Algorithm with “fibonacci growth”.
- No idea to enlarge this growth.

Construction of a lower bound:
- Start with an arbitrary algorithm.
- Use only the restriction of the algorithm.
- Abstract.

We will now try to do the abstraction.

Try the get the core-problem.

The core-problem ist:
- “Fibonacci growth” could not be improved.
Idea for the lower Bound

- **Situation:**
  - Algorithm with “fibonacci growth”.
  - No idea to enlarge this growth.

- **Construction of a lower bound:**
  - Start with an arbitrary algorithm.
  - Use only the restriction of the algorithm.
  - Abstract.

- We will now try to do the abstraction.

- Try the get the core-problem.

- The core-problem ist:
  - “Fibonacci growth” could not be improved.
1. Abstraction

**Definition:**

The **Network Counting Problem**:

- Given a directed graph $G = (V, E)$.
- Each node stores a number.
- Initial just the number 1 is stored.
- The receiver add the number from the sender to his number after one communication.
- The objective is: all nodes should store a number larger then $|V|$.
- With $nc(G)$ we denote the minimal rounds to achieve this objective.

**Lemma:**

For any graph $G$ we have: $r(G) \geq nc(G)$. 
1. Abstraction

**Definition:**

The Network Counting Problem:

- Given a directed graph $G = (V, E)$.
- Each node stores a number.
- Initial just the number 1 is stored.
- The receiver adds the number from the sender to his number after one communication.
- The objective is: all nodes should store a number larger than $|V|$.
- With $nc(G)$ we denote the minimal rounds to achieve this objective.

**Lemma:**

For any graph $G$ we have: $r(G) \geq nc(G)$. 
1. Abstraction

**Definition:**

The **Network Counting Problem**:

- Given a directed graph $G = (V, E)$.
- Each node stores a number.
- **Initial just the number 1 is stored.**
- The receiver add the number from the sender to his number after one communication.
- The objective is: all nodes should store a number larger then $|V|$.
- With $nc(G)$ we denote the minimal rounds to achieve this objective.

**Lemma:**

For any graph $G$ we have: $r(G) \geq nc(G)$. 
1. Abstraction

**Definition:**

The Network Counting Problem:

- Given a directed graph $G = (V, E)$.
- Each node stores a number.
- Initial just the number 1 is stored.
- The receiver add the number from the sender to his number after one communication.
- The objective is: all nodes should store a number larger then $|V|$.
- With $nc(G)$ we denote the minimal rounds to achieve this objective.

**Lemma:**

For any graph $G$ we have: $r(G) \geq nc(G)$. 
1. Abstraction

**Definition:**

The Network Counting Problem:
- Given a directed graph $G = (V, E)$.
- Each node stores a number.
- Initial just the number 1 is stored.
- The receiver add the number from the sender to his number after one communication.
- The objective is: all nodes should store a number larger then $|V|$.
- With $nc(G)$ we denote the minimal rounds to achieve this objective.

**Lemma:**

For any graph $G$ we have: $r(G) \geq nc(G)$. 
1. Abstraction

**Definition:**

The Network Counting Problem:

- Given a directed graph $G = (V, E)$.
- Each node stores a number.
- Initial just the number 1 is stored.
- The receiver add the number from the sender to his number after one communication.
- The objective is: all nodes should store a number larger than $|V|$.
- With $nc(G)$ we denote the minimal rounds to achieve this objective.

**Lemma:**

For any graph $G$ we have: $r(G) \geq nc(G)$. 
1. Abstraction

**Definition:**

The **Network Counting Problem**:
- Given a directed graph $G = (V, E)$.
- Each node stores a number.
- Initial just the number 1 is stored.
- The receiver add the number from the sender to his number after one communication.
- The objective is: all nodes should store a number larger than $|V|$.
- With $nc(G)$ we denote the minimal rounds to achieve this objective.

**Lemma:**

For any graph $G$ we have: $r(G) \geq nc(G)$. 
1. Abstraction

**Definition:**

The **Network Counting Problem**:

- Given a directed graph $G = (V, E)$.
- Each node stores a number.
- Initial just the number 1 is stored.
- The receiver adds the number from the sender to his number after one communication.
- The objective is: all nodes should store a number larger than $|V|$.

With $nc(G)$ we denote the minimal rounds to achieve this objective.

**Lemma:**

For any graph $G$ we have: $r(G) \geq nc(G)$. 
2. Abstraction

- Let $G = (\{v_1, v_2, v_3, \ldots, v_n\}, E)$ be a directed Graph.
- Each node $v_i$ stores after $t$ rounds the number $z_i^t$.
- One situation of the network counting problem could be described by a vector:
  - Initial: $(1, 1, 1, \ldots, 1)^T$.
  - After $t$ rounds: $(z_1^t, z_2^t, z_3^t, z_n^t)^T$.

- One round of an algorithm for the network counting problem is given by a matrix $B$:
  - $A$ is a $n \times n$ matrix.
  - $a_{ij} = 1$ node $j$ sends to node $i$.
  - $A$ contains on the diagonal only ones.
  - $A$ has in each row at most two ones.
  - $A$ has in each column at most two ones.
  - If $a_{ij} = a_{kl} = 1$ ($i \neq j \neq k \neq l$), then we have $l \neq i \neq k$ and $l \neq j \neq k$.
  - Thus we get: $A \cdot (z_1^t, z_2^t, z_3^t, z_n^t)^T = (z_1^{t+1}, z_2^{t+1}, z_3^{t+1}, z_n^{t+1})^T$.
2. Abstraction

- Let $G = (\{v_1, v_2, v_3, \cdots, v_n\}, E)$ be a directed Graph.
- Each node $v_i$ stores after $t$ rounds the number $z_i^t$.

One situation of the network counting problem could be described by a vector:

- Initial: $(1, 1, 1, \cdots, 1)^T$.
- After $t$ rounds: $(z_1^t, z_2^t, z_3^t, \ldots, z_n^t)^T$.

One round of an algorithm for the network counting problem is given by a matrix $B$:

- $A$ is a $n \times n$ matrix.
- $a_{ij} = 1$ node $j$ sends to node $i$.
- $A$ contains on the diagonal only ones.
- $A$ has in each row at most two ones.
- $A$ has in each column at most two ones.
- If $a_{ij} = a_{kl} = 1$ ($i \neq j \neq k \neq l$), then we have $l \neq i \neq k$ and $l \neq j \neq k$.
- Thus we get: $A \cdot (z_1^t, z_2^t, z_3^t, z_n^t)^T = (z_1^{t+1}, z_2^{t+1}, z_3^{t+1}, z_n^{t+1})^T$
2. Abstraction

- Let $G = (\{v_1, v_2, v_3, \ldots, v_n\}, E)$ be a directed Graph.
- Each node $v_i$ stores after $t$ rounds the number $z_i^t$.
- One situation of the network counting problem could be described by a vector:
  - Initial: $(1, 1, 1, \ldots, 1)^T$.
  - After $t$ rounds: $(z_1^t, z_2^t, z_3^t, z_n^t)^T$.
- One round of an algorithm for the network counting problem is given by a matrix $B$:
  - $A$ is a $n \times n$ matrix.
  - $a_{ij} = 1$ node $j$ sends to node $i$.
  - $A$ contains on the diagonal only ones.
  - $A$ has in each row at most two ones.
  - $A$ has in each column at most two ones.
  - If $a_{ij} = a_{kl} = 1$ ($i \neq j \neq k \neq l$), then we have $l \neq i \neq k$ and $l \neq j \neq k$.
  - Thus we get: $A \cdot (z_1^t, z_2^t, z_3^t, z_n^t)^T = (z_1^{t+1}, z_2^{t+1}, z_3^{t+1}, z_n^{t+1})^T$.
2. Abstraction

- Let $G = (\{v_1, v_2, v_3, \cdots, v_n\}, E)$ be a directed Graph.
- Each node $v_i$ stores after $t$ rounds the number $z_i^t$.
- One situation of the network counting problem could be described by a vector:
  - Initial: $(1,1,1,\cdots,1)^T$.
  - After $t$ rounds: $(z_1^t, z_2^t, z_3^t, z_n^t)^T$.
- One round of an algorithm for the network counting problem is given by a matrix $B$:
  - $A$ is a $n \times n$ matrix.
  - $a_{ij} = 1$ node $j$ sends to node $i$.
  - $A$ contains on the diagonal only ones.
  - $A$ has in each row at most two ones.
  - $A$ has in each column at most two ones.
  - If $a_{ij} = a_{kl} = 1$ ($i \neq j \neq k \neq l$), then we have $l \neq i \neq k$ and $l \neq j \neq k$.
  - Thus we get: $A \cdot (z_1^t, z_2^t, z_3^t, z_n^t)^T = (z_1^{t+1}, z_2^{t+1}, z_3^{t+1}, z_n^{t+1})^T$.
2. Abstraction

- Let $G = (\{v_1, v_2, v_3, \cdots, v_n\}, E)$ be a directed Graph.
- Each node $v_i$ stores after $t$ rounds the number $z_i^t$.
- One situation of the network counting problem could be described by a vector:
  - Initial: $(1, 1, 1, \cdots, 1)^T$.
  - After $t$ rounds: $(z_1^t, z_2^t, z_3^t, z_n^t)^T$.
- One round of an algorithm for the network counting problem is given by a matrix $B$:
  - $A$ is a $n \times n$ matrix.
  - $a_{ij} = 1$ node $j$ sends to node $i$.
  - $A$ contains on the diagonal only ones.
  - $A$ has in each row at most two ones.
  - $A$ has in each column at most two ones.
  - If $a_{ij} = a_{kl} = 1$ ($i \neq j \neq k \neq l$), then we have $l \neq i \neq k$ and $l \neq j \neq k$.
  - Thus we get: $A \cdot (z_1^t, z_2^t, z_3^t, z_n^t)^T = (z_1^{t+1}, z_2^{t+1}, z_3^{t+1}, z_n^{t+1})^T$.
2. Abstraction

- Let $G = (\{v_1, v_2, v_3, \cdots, v_n\}, E)$ be a directed Graph.
- Each node $v_i$ stores after $t$ rounds the number $z_{i}^{t}$.
- One situation of the network counting problem could be described by a vector:
  - Initial: $(1, 1, 1, \cdots, 1)^{T}$.
  - After $t$ rounds: $(z_{1}^{t}, z_{2}^{t}, z_{3}^{t}, \cdots, z_{n}^{t})^{T}$.
- One round of an algorithm for the network counting problem is given by a matrix $B$:
  - $A$ is a $n \times n$ matrix.
  - $a_{ij} = 1$ node $j$ sends to node $i$.
  - $A$ contains on the diagonal only ones.
  - $A$ has in each row at most two ones.
  - $A$ has in each column at most two ones.
  - If $a_{ij} = a_{kl} = 1$ ($i \neq j \neq k \neq l$), then we have $l \neq i \neq k$ and $l \neq j \neq k$.
  - Thus we get: $A \cdot (z_{1}^{t}, z_{2}^{t}, z_{3}^{t}, \cdots, z_{n}^{t})^{T} = (z_{1}^{t+1}, z_{2}^{t+1}, z_{3}^{t+1}, \cdots, z_{n}^{t+1})^{T}$.
2. Abstraction

- Let $G = (\{v_1, v_2, v_3, \cdots, v_n\}, E)$ be a directed Graph.
- Each node $v_i$ stores after $t$ rounds the number $z_i^t$.
- One situation of the network counting problem could be described by a vector:
  - Initial: $(1, 1, 1, \cdots, 1)^T$.
  - After $t$ rounds: $(z_1^t, z_2^t, z_3^t, z_n^t)^T$.
- One round of an algorithm for the network counting problem is given by a matrix $B$:
  - $A$ is a $n \times n$ matrix.
  - $a_{ij} = 1$ node $j$ sends to node $i$.
  - $A$ contains on the diagonal only ones.
  - $A$ has in each row at most two ones.
  - $A$ has in each column at most two ones.
  - If $a_{ij} = a_{kl} = 1$ ($i \neq j \neq k \neq l$), then we have $l \neq i \neq k$ and $l \neq j \neq k$.
  - Thus we get: $A \cdot (z_1^t, z_2^t, z_3^t, z_n^t)^T = (z_1^{t+1}, z_2^{t+1}, z_3^{t+1}, z_n^{t+1})^T$.
2. Abstraction

- Let $G = (\{v_1, v_2, v_3, \cdots, v_n\}, E)$ be a directed Graph.
- Each node $v_i$ stores after $t$ rounds the number $z_i^t$.
- One situation of the network counting problem could be described by a vector:
  - Initial: $(1, 1, 1, \cdots, 1)^T$.
  - After $t$ rounds: $(z_1^t, z_2^t, z_3^t, z_n^t)^T$.
- One round of an algorithm for the network counting problem is given by a matrix $B$:
  - $A$ is a $n \times n$ matrix.
  - $a_{ij} = 1$ node $j$ sends to node $i$.
  - $A$ contains on the diagonal only ones.
  - $A$ has in each row at most two ones.
  - $A$ has in each column at most two ones.
  - If $a_{ij} = a_{kl} = 1$ ($i \neq j \neq k \neq l$), then we have $l \neq i \neq k$ and $l \neq j \neq k$.
  - Thus we get: $A \cdot (z_1^t, z_2^t, z_3^t, z_n^t)^T = (z_1^{t+1}, z_2^{t+1}, z_3^{t+1}, z_n^{t+1})^T$. 

2. Abstraction

- Let $G = (\{v_1, v_2, v_3, \ldots, v_n\}, E)$ be a directed Graph.
- Each node $v_i$ stores after $t$ rounds the number $z_i^t$.
- One situation of the network counting problem could be described by a vector:
  - Initial: $(1, 1, 1, \ldots, 1)^T$.
  - After $t$ rounds: $(z_1^t, z_2^t, z_3^t, z_n^t)^T$.
- One round of an algorithm for the network counting problem is given by a matrix $B$:
  - $A$ is a $n \times n$ matrix.
  - $a_{ij} = 1$ node $j$ sends to node $i$.
  - $A$ contains on the diagonal only ones.
  - $A$ has in each row at most two ones.
  - $A$ has in each column at most two ones.
  - If $a_{ij} = a_{kl} = 1$ ($i \neq j \neq k \neq l$), then we have $l \neq i \neq k$ and $l \neq j \neq k$.
  - Thus we get: $A \cdot (z_1^t, z_2^t, z_3^t, z_n^t)^T = (z_1^{t+1}, z_2^{t+1}, z_3^{t+1}, z_n^{t+1})^T$.
2. Abstraction

- Let $G = (\{v_1, v_2, v_3, \cdots, v_n\}, E)$ be a directed Graph.
- Each node $v_i$ stores after $t$ rounds the number $z_i^t$.
- One situation of the network counting problem could be described by a vector:
  - Initial: $(1, 1, 1, \cdots, 1)^T$.
  - After $t$ rounds: $(z_1^t, z_2^t, z_3^t, z_n^t)^T$.
- One round of an algorithm for the network counting problem is given by a matrix $B$:
  - $A$ is a $n \times n$ matrix.
  - $a_{ij} = 1$ node $j$ sends to node $i$.
  - $A$ contains on the diagonal only ones.
  - $A$ has in each row at most two ones.
  - $A$ has in each column at most two ones.
  - If $a_{ij} = a_{kl} = 1$ ($i \neq j \neq k \neq l$), then we have $l \neq i \neq k$ and $l \neq j \neq k$.
  - Thus we get: $A \cdot (z_1^t, z_2^t, z_3^t, z_n^t)^T = (z_1^{t+1}, z_2^{t+1}, z_3^{t+1}, z_n^{t+1})^T$.
2. Abstraction

- Let $G = (\{v_1, v_2, v_3, \cdots, v_n\}, E)$ be a directed Graph.
- Each node $v_i$ stores after $t$ rounds the number $z_i^t$.
- One situation of the network counting problem could be described by a vector:
  - Initial: $(1, 1, 1, \cdots, 1)^T$.
  - After $t$ rounds: $(z_1^t, z_2^t, z_3^t, z_n^t)^T$.
- One round of an algorithm for the network counting problem is given by a matrix $B$:
  - $A$ is a $n \times n$ matrix.
  - $a_{ij} = 1$ node $j$ sends to node $i$.
  - $A$ contains on the diagonal only ones.
  - $A$ has in each row at most two ones.
  - $A$ has in each column at most two ones.
  - If $a_{ij} = a_{kl} = 1$ ($i \neq j \neq k \neq l$), then we have $l \neq i \neq k$ and $l \neq j \neq k$.
  - Thus we get: $A \cdot (z_1^t, z_2^t, z_3^t, z_n^t)^T = (z_1^{t+1}, z_2^{t+1}, z_3^{t+1}, z_n^{t+1})^T$.
2. Abstraction

- Let \( G = (\{v_1, v_2, v_3, \cdots, v_n\}, E) \) be a directed Graph.
- Each node \( v_i \) stores after \( t \) rounds the number \( z_i^t \).
- One situation of the network counting problem could be described by a vector:
  - Initial: \((1, 1, 1, \cdots, 1)^T\).
  - After \( t \) rounds: \((z_1^t, z_2^t, z_3^t, z_n^t)^T\).
- One round of an algorithm for the network counting problem is given by a matrix \( B \):
  - \( A \) is a \( n \times n \) matrix.
  - \( a_{ij} = 1 \) node \( j \) sends to node \( i \).
  - \( A \) contains on the diagonal only ones.
  - \( A \) has in each row at most two ones.
  - \( A \) has in each column at most two ones.
  - If \( a_{ij} = a_{kl} = 1 \) \((i \neq j \neq k \neq l)\), then we have \( l \neq i \neq k \) and \( l \neq j \neq k \).
  - Thus we get: \( A \cdot (z_1^t, z_2^t, z_3^t, z_n^t)^T = (z_1^{t+1}, z_2^{t+1}, z_3^{t+1}, z_n^{t+1})^T \)
2. Abstraction

Let $G = (\{v_1, v_2, v_3, \cdots, v_n\}, E)$ be a directed Graph.

Each node $v_i$ stores after $t$ rounds the number $z_i^t$.

One situation of the network counting problem could be described by a vector:

- Initial: $(1, 1, 1, \cdots, 1)^T$.
- After $t$ rounds: $(z_1^t, z_2^t, z_3^t, z_n^t)^T$.

One round of an algorithm for the network counting problem is given by a matrix $B$:

- $A$ is a $n \times n$ matrix.
- $a_{ij} = 1$ node $j$ sends to node $i$.
- $A$ contains on the diagonal only ones.
- $A$ has in each row at most two ones.
- $A$ has in each column at most two ones.
- If $a_{ij} = a_{kl} = 1$ ($i \neq j \neq k \neq l$), then we have $l \neq i \neq k$ and $l \neq j \neq k$.
- Thus we get: $A \cdot (z_1^t, z_2^t, z_3^t, z_n^t)^T = (z_1^{t+1}, z_2^{t+1}, z_3^{t+1}, z_n^{t+1})^T$. 

2. Abstraction (Continuation)

- We consider now matrices of the above form.
- These are matrices $A$, for which there is a transformation $T$ with:

$$TAT^{-1} = \begin{pmatrix} B & 0 \\ \cdot & \cdot \\ \cdot & B \\ 0 & 1 \end{pmatrix}.$$ 

and $B = \begin{pmatrix} 11 \\ 01 \end{pmatrix}$.

- We will estimate the growth, which these matrices provide for the network counting problem.
2. Abstraction (Continuation)

- We consider now matrices of the above form.
- These are matrices $A$, for which there is a transformation $T$ with:

$$TAT^{-1} = \begin{pmatrix} B & & & 0 \\ & B & & \\ & & \ddots & B \\ 0 & & & 1 \\ & & & \ddots \\ & & & & 1 \end{pmatrix}.$$ 

and $B = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$.

- We will estimate the growth, which these matrices provide for the network counting problem.
2. Abstraction (Continuation)

We consider now matrices of the above form. These are matrices $A$, for which there is a transformation $T$ with:

$$TAT^{-1} = \begin{pmatrix} B & \ddots & \ddots & 0 \\ \ddots & B & \ddots & 1 \\ \ddots & \ddots & B & \ddots \\ 0 & \ddots & \ddots & 1 \end{pmatrix}.$$ 

and $B = \begin{pmatrix} 11 \\ 01 \end{pmatrix}$.

We will estimate the growth, which these matrices provide for the network counting problem.
Recollection (Norm, 3. Abstraction)

Let $||\cdot||$ be the vector norm over $\mathbb{R}^n$. Then we have:

- $||x|| = 0 \Leftrightarrow x = 0^n$,
- $||\alpha \cdot x|| = |\alpha| \cdot ||x||$,
- $||x + y|| \leq ||x|| + ||y||$
- this holds for all $\alpha \in \mathbb{R}$, $x, y \in \mathbb{R}^n$

The matrix norm for a vector norm $||\cdot||$ is defined by $||A|| = \sup_{x \neq 0} \frac{||Ax||}{||x||}$.

Then we have:

- $||A|| = 0 \Leftrightarrow A = 0$
- $||A + B|| \leq ||A|| + ||B||$
- $||\alpha A|| = |\alpha| \cdot ||A||$
- $||A \cdot B|| \leq ||A|| \cdot ||B||$
- $||A \cdot x|| \leq ||A|| \cdot ||x||$
- this holds for all $A, B \in \mathbb{R}^{n^2}, x \in \mathbb{R}^n, \alpha \in \mathbb{R}, \alpha \geq 0$.

Here we use: $||x|| = \sqrt{\sum_{i=1}^{n} |x_i|^2}$ for $x = (x_1, \ldots, x_n)$,

Known: $||A|| = \text{Spectral Norm}(A) = \sqrt{\lambda_{\text{max}}(A^T \cdot A)}$ with: $\lambda_{\text{max}}$ is the largest Eigenvalue.
Recollection (Norm, 3. Abstraction)

- Let $||.||$ be the vector norm over $\mathbb{R}^n$. Then we have:
  - $||x|| = 0 \iff x = 0^n$,
  - $||\alpha \cdot x|| = |\alpha| \cdot ||x||$,
  - $||x + y|| \leq ||x|| + ||y||$
  - this holds for all $\alpha \in \mathbb{R}, x, y \in \mathbb{R}^n$

- The matrix norm for a vector norm $||.||$ is defined by $||A|| = \sup_{x \neq 0} \frac{||Ax||}{||x||}$. Then we have:
  - $||A|| = 0 \iff A = 0$
  - $||A + B|| \leq ||A|| + ||B||$
  - $||\alpha A|| = \alpha \cdot ||A||$
  - $||A \cdot B|| \leq ||A|| \cdot ||B||$
  - $||A \cdot x|| \leq ||A|| \cdot ||x||$
  - this holds for all $A, B \in \mathbb{R}^{n^2}, x \in \mathbb{R}^n, \alpha \in \mathbb{R}, \alpha \geq 0$.

- Here we use: $||x|| = \sqrt{\sum_{i=1}^{n}|x_i|^2}$ for $x = (x_1, .., x_n)$,

- Known: $||A|| = \text{Spectral Norm}(A) = \sqrt{\lambda_{max}(A^T \cdot A)}$ with: $\lambda_{max}$ is the largest Eigenvalue.
Recollection (Norm, 3. Abstraction)

- Let \(|\cdot|\) be the vector norm over \(\mathbb{R}^n\). Then we have:
  - \(|x| = 0 \iff x = 0^n\),
  - \(|\alpha \cdot x| = |\alpha| \cdot |x|\),
  - \(|x + y| \leq |x| + |y|\)
  - this holds for all \(\alpha \in \mathbb{R}, x, y \in \mathbb{R}^n\)

- The matrix norm for a vector norm \(|\cdot|\) is defined by \(|A| = \sup_{x \neq 0} \frac{|Ax|}{|x|}\). Then we have:
  - \(|A| = 0 \iff A = 0\)
  - \(|A + B| \leq |A| + |B|\)
  - \(|\alpha A| = |\alpha| \cdot |A|\)
  - \(|A \cdot B| \leq |A| \cdot |B|\)
  - \(|A \cdot x| \leq |A| \cdot |x|\)
  - this holds for all \(A, B \in \mathbb{R}^{n^2}, x \in \mathbb{R}^n, \alpha \in \mathbb{R}, \alpha \geq 0\).

- Here we use: \(|x| = \sqrt{\sum_{i=1}^{n} |x_i|^2}\) for ein \(x = (x_1, .., x_n)\),

- Known: \(|A| = \text{Spectral Norm}(A) = \sqrt{\lambda_{max}(A^T \cdot A)}\) with: \(\lambda_{max}\) is the largest Eigenvalue.
Recollection (Norm, 3. Abstraction)

- Let $\|..\|$ be the vector norm over $\mathbb{R}^n$. Then we have:
  - $\|x\| = 0 \iff x = 0^n$,
  - $\|\alpha \cdot x\| = |\alpha| \cdot \|x\|$, 
  - $\|x + y\| \leq \|x\| + \|y\|$ 
  - this holds for all $\alpha \in \mathbb{R}, x, y \in \mathbb{R}^n$

- The matrix norm for a vector norm $\|..\|$ is defined by $\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|}$. Then we have:
  - $\|A\| = 0 \iff A = 0$
  - $\|A + B\| \leq \|A\| + \|B\|$
  - $\|\alpha A\| = |\alpha| \cdot \|A\|$
  - $\|A \cdot B\| \leq \|A\| \cdot \|B\|$
  - $\|A \cdot x\| \leq \|A\| \cdot \|x\|$
  - this holds for all $A, B \in \mathbb{R}^{n^2}, x \in \mathbb{R}^n, \alpha \in \mathbb{R}, \alpha \geq 0$.

- Here we use: $\|x\| = \sqrt{\sum_{i=1}^n |x_i|^2}$ for ein $x = (x_1, .., x_n)$,

- Known: $\|A\| = \text{Spectral Norm}(A) = \sqrt{|\lambda_{\text{max}}(A^T \cdot A)|}$ with: $\lambda_{\text{max}}$ is the largest Eigenvalue.
Recollection (Norm, 3. Abstraction)

- Let $||\cdot||$ be the vector norm over $\mathbb{R}^n$. Then we have:
  - $||x|| = 0 \iff x = 0^n$,
  - $||\alpha \cdot x|| = |\alpha| \cdot ||x||$,
  - $||x + y|| \leq ||x|| + ||y||$
  - this holds for all $\alpha \in \mathbb{R}, x, y \in \mathbb{R}^n$

- The matrix norm for a vector norm $||\cdot||$ is defined by $||A|| = \sup_{x \neq 0} \frac{||Ax||}{||x||}$. Then we have:
  - $||A|| = 0 \iff A = 0$
  - $||A + B|| \leq ||A|| + ||B||$
  - $||\alpha A|| = |\alpha| \cdot ||A||$
  - $||A \cdot B|| \leq ||A|| \cdot ||B||$
  - $||A \cdot x|| \leq ||A|| \cdot ||x||$
  - this holds for all $A, B \in \mathbb{R}^{n^2}, x \in \mathbb{R}^n, \alpha \in \mathbb{R}, \alpha \geq 0$.

- Here we use: $||x|| = \sqrt{\sum_{i=1}^{n} |x_i|^2}$ for $x = (x_1, \ldots, x_n)$,

- Known: $||A|| = \text{Spectral Norm}(A) = \sqrt{|\lambda_{\text{max}}(A^T \cdot A)|}$ with: $\lambda_{\text{max}}$ is the largest Eigenvalue.
Recollection (Norm, 3. Abstraction)

Let $||.||$ be the vector norm over $\mathbb{R}^n$. Then we have:

- $||x|| = 0 \iff x = 0^n$,
- $||\alpha \cdot x|| = |\alpha| \cdot ||x||$,
- $||x + y|| \leq ||x|| + ||y||$
- this holds for all $\alpha \in \mathbb{R}, x, y \in \mathbb{R}^n$

The matrix norm for a vector norm $||.||$ is defined by $||A|| = \sup_{x \neq 0} \frac{||Ax||}{||x||}$. Then we have:

- $||A|| = 0 \iff A = 0$
- $||A + B|| \leq ||A|| + ||B||$
- $||\alpha A|| = |\alpha| \cdot ||A||$
- $||A \cdot B|| \leq ||A|| \cdot ||B||$
- $||A \cdot x|| \leq ||A|| \cdot ||x||$
- this holds for all $A, B \in \mathbb{R}^{n^2}, x \in \mathbb{R}^n, \alpha \in \mathbb{R}, \alpha \geq 0$.

Here we use: $||x|| = \sqrt{\sum_{i=1}^{n} |x_i|^2}$ for ein $x = (x_1, .., x_n)$,

Known: $||A|| = \text{Spectral Norm}(A) = \sqrt{|\lambda_{max}(A^T \cdot A)|}$ with: $\lambda_{max}$ is the largest Eigenvalue.
Recollection (Norm, 3. Abstraction)

- Let $||.||$ be the vector norm over $\mathbb{R}^n$. Then we have:
  - $||x|| = 0 \iff x = 0^n$,
  - $||\alpha \cdot x|| = |\alpha| \cdot ||x||$,
  - $||x + y|| \leq ||x|| + ||y||$
  - this holds for all $\alpha \in \mathbb{R}, x, y \in \mathbb{R}^n$

- The matrix norm for a vector norm $||.||$ is defined by $||A|| = sup_{x \neq 0} \frac{||Ax||}{||x||}$. Then we have:
  - $||A|| = 0 \iff A = 0$
  - $||A + B|| \leq ||A|| + ||B||$
  - $||\alpha A|| = |\alpha| \cdot ||A||$
  - $||A \cdot B|| \leq ||A|| \cdot ||B||$
  - $||A \cdot x|| \leq ||A|| \cdot ||x||$
  - this holds for all $A, B \in \mathbb{R}^{n^2}, x \in \mathbb{R}^n, \alpha \in \mathbb{R}, \alpha \geq 0$.

- Here we use: $||x|| = \sqrt{\sum_{i=1}^{n} |x_i|^2}$ for $\text{ein } x = (x_1, \ldots, x_n)$,

- Known: $||A|| = \text{ Spectral Norm}(A) = \sqrt{\lambda_{max}(A^T \cdot A)}$ with: $\lambda_{max}$ is the largest Eigenvalue.
Recollection (Norm, 3. Abstraction)

- Let $||.||$ be the vector norm over $\mathbb{R}^n$. Then we have:
  - $||x|| = 0 \iff x = 0^n$,
  - $||\alpha \cdot x|| = |\alpha| \cdot ||x||$,
  - $||x + y|| \leq ||x|| + ||y||$
  - this holds for all $\alpha \in \mathbb{R}, x, y \in \mathbb{R}^n$

- The matrix norm for a vector norm $||.||$ is defined by $||A|| = \sup_{x \neq 0} \frac{||Ax||}{||x||}$. Then we have:
  - $||A|| = 0 \iff A = 0$
  - $||A + B|| \leq ||A|| + ||B||$
  - $||\alpha A|| = \alpha \cdot ||A||$
  - $||A \cdot B|| \leq ||A|| \cdot ||B||$
  - $||A \cdot x|| \leq ||A|| \cdot ||x||$
  - this holds for all $A, B \in \mathbb{R}^{n^2}, x \in \mathbb{R}^n, \alpha \in \mathbb{R}, \alpha \geq 0$.

- Here we use: $||x|| = \sqrt{\sum_{i=1}^{n} |x_i|^2}$ for ein $x = (x_1, .., x_n)$,

- Known: $||A|| = \text{Spectral Norm}(A) = \sqrt{|\lambda_{max}(A^T \cdot A)|}$ with: $\lambda_{max}$ is the largest Eigenvalue.
Recollection (Norm, 3. Abstraction)

- Let $||..||$ be the vector norm over $\mathbb{R}^n$. Then we have:
  - $||x|| = 0 \iff x = 0^n$,
  - $||\alpha \cdot x|| = |\alpha| \cdot ||x||$,
  - $||x + y|| \leq ||x|| + ||y||$
  - this holds for all $\alpha \in \mathbb{R}, x, y \in \mathbb{R}^n$

- The matrix norm for a vector norm $||..||$ is defined by $||A|| = sup_{x \neq 0} \frac{||Ax||}{||x||}$. Then we have:
  - $||A|| = 0 \iff A = 0$
  - $||A + B|| \leq ||A|| + ||B||$
  - $||\alpha A|| = \alpha \cdot ||A||$
  - $||A \cdot B|| \leq ||A|| \cdot ||B||$
  - $||A \cdot x|| \leq ||A|| \cdot ||x||$
  - this holds for all $A, B \in \mathbb{R}^{n^2}, x \in \mathbb{R}^n, \alpha \in \mathbb{R}, \alpha \geq 0$.

- Here we use: $||x|| = \sqrt{\sum_{i=1}^{n} |x_i|^2}$ for ein $x = (x_1, .., x_n)$,

- Known: $||A|| = \text{Spectral Norm}(A) = \sqrt{|\lambda_{max}(A^T \cdot A)|}$ with: $\lambda_{max}$ is the largest Eigenvalue.
Let $||.||$ be the vector norm over $\mathbb{R}^n$. Then we have:

- $||x|| = 0 \iff x = 0^n$,
- $||\alpha \cdot x|| = |\alpha| \cdot ||x||$,
- $||x + y|| \leq ||x|| + ||y||$
- this holds for all $\alpha \in \mathbb{R}, x, y \in \mathbb{R}^n$

The matrix norm for a vector norm $||.||$ is defined by $||A|| = \sup_{x \neq 0}\frac{||Ax||}{||x||}$. Then we have:

- $||A|| = 0 \iff A = 0$
- $||A + B|| \leq ||A|| + ||B||$
- $||\alpha A|| = \alpha \cdot ||A||$
- $||A \cdot B|| \leq ||A|| \cdot ||B||$
- $||A \cdot x|| \leq ||A|| \cdot ||x||$
- this holds for all $A, B \in \mathbb{R}^{n^2}, x \in \mathbb{R}^n, \alpha \in \mathbb{R}, \alpha \geq 0$.

Here we use: $||x|| = \sqrt{\sum_{i=1}^{n} |x_i|^2}$ for ein $x = (x_1, ..., x_n)$,

Known: $||A|| = \text{Spectral Norm}(A) = \sqrt{|\lambda_{max}(A^T \cdot A)|}$ with: $\lambda_{max}$ is the largest Eigenvalue.
Recollection (Norm, 3. Abstraction)

Let $||\cdot||$ be the vector norm over $\mathbb{R}^n$. Then we have:

- $||x|| = 0 \iff x = 0^n$,
- $||\alpha \cdot x|| = |\alpha| \cdot ||x||$,
- $||x + y|| \leq ||x|| + ||y||$,
- this holds for all $\alpha \in \mathbb{R}, x, y \in \mathbb{R}^n$.

The matrix norm for a vector norm $||\cdot||$ is defined by $||A|| = \sup_{x \neq 0} \frac{||Ax||}{||x||}$.

Then we have:

- $||A|| = 0 \iff A = 0$,
- $||A + B|| \leq ||A|| + ||B||$,
- $||\alpha A|| = |\alpha| \cdot ||A||$,
- $||A \cdot B|| \leq ||A|| \cdot ||B||$,
- $||A \cdot x|| \leq ||A|| \cdot ||x||$,
- this holds for all $A, B \in \mathbb{R}^{n^2}, x \in \mathbb{R}^n, \alpha \in \mathbb{R}, \alpha \geq 0$.

Here we use: $||x|| = \sqrt{\sum_{i=1}^{n} |x_i|^2}$ for $e_{in} x = (x_1, \ldots, x_n)$.

Known: $||A|| = \text{Spectral Norm}(A) = \sqrt{|\lambda_{\text{max}}(A^T \cdot A)|}$ with: $\lambda_{\text{max}}$ is the largest Eigenvalue.
Let $\|\cdot\|$ be the vector norm over $\mathbb{R}^n$. Then we have:

- $\|x\| = 0 \iff x = 0^n$,
- $\|\alpha \cdot x\| = |\alpha| \cdot \|x\|$,
- $\|x + y\| \leq \|x\| + \|y\|$
- this holds for all $\alpha \in \mathbb{R}, x, y \in \mathbb{R}^n$

The matrix norm for a vector norm $\|\cdot\|$ is defined by $\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|}$. Then we have:

- $\|A\| = 0 \iff A = 0$
- $\|A + B\| \leq \|A\| + \|B\|$
- $\|\alpha A\| = \alpha \cdot \|A\|$
- $\|A \cdot B\| \leq \|A\| \cdot \|B\|$
- $\|A \cdot x\| \leq \|A\| \cdot \|x\|$
- this holds for all $A, B \in \mathbb{R}^{n \times n}, x \in \mathbb{R}^n, \alpha \in \mathbb{R}, \alpha \geq 0.$

Here we use: $\|x\| = \sqrt{\sum_{i=1}^{n} |x_i|^2}$ for $e^{i \theta} = (x_1, \ldots, x_n)$,

Known: $\|A\| = \text{Spectral Norm}(A) = \sqrt{|\lambda_{\max}(A^T \cdot A)|}$ with: $\lambda_{\max}$ is the largest Eigenvalue.
Recollection (Norm, 3. Abstraction)

Let $||\cdot||$ be the vector norm over $\mathbb{R}^n$. Then we have:

- $||x|| = 0 \iff x = 0^n$,
- $||\alpha \cdot x|| = |\alpha| \cdot ||x||$,
- $||x + y|| \leq ||x|| + ||y||$
- this holds for all $\alpha \in \mathbb{R}, x, y \in \mathbb{R}^n$

The matrix norm for a vector norm $||\cdot||$ is defined by $||A|| = \sup_{x \neq 0} \frac{||Ax||}{||x||}$. Then we have:

- $||A|| = 0 \iff A = 0$
- $||A + B|| \leq ||A|| + ||B||$
- $||\alpha A|| = \alpha \cdot ||A||$
- $||A \cdot B|| \leq ||A|| \cdot ||B||$
- $||A \cdot x|| \leq ||A|| \cdot ||x||$
- this holds for all $A, B \in \mathbb{R}^{n^2}, x \in \mathbb{R}^n, \alpha \in \mathbb{R}, \alpha \geq 0$.

Here we use: $||x|| = \sqrt{\sum_{i=1}^{n} |x_i|^2}$ for $x = (x_1, \ldots, x_n)$,

Known: $||A|| = \text{Spectral Norm}(A) = \sqrt{|\lambda_{\text{max}}(A^T \cdot A)|}$ with: $\lambda_{\text{max}}$ is the largest Eigenvalue.
Recollection (Norm, 3. Abstraction)

Let $||\cdot||$ be the vector norm over $\mathbb{R}^n$. Then we have:

- $||x|| = 0 \iff x = 0^n$,
- $||\alpha \cdot x|| = |\alpha| \cdot ||x||$,
- $||x + y|| \leq ||x|| + ||y||$
- this holds for all $\alpha \in \mathbb{R}, x, y \in \mathbb{R}^n$

The matrix norm for a vector norm $||\cdot||$ is defined by $||A|| = \sup_{x \neq 0} \frac{||Ax||}{||x||}$. Then we have:

- $||A|| = 0 \iff A = 0$
- $||A + B|| \leq ||A|| + ||B||$
- $||\alpha A|| = |\alpha| \cdot ||A||$
- $||A \cdot B|| \leq ||A|| \cdot ||B||$
- $||A \cdot x|| \leq ||A|| \cdot ||x||$
- this holds for all $A, B \in \mathbb{R}^{n^2}, x \in \mathbb{R}^n, \alpha \in \mathbb{R}, \alpha \geq 0$.

Here we use: $||x|| = \sqrt{\sum_{i=1}^{n} |x_i|^2}$ for $x = (x_1, .., x_n)$,

Known: $||A|| = \text{Spectral Norm}(A) = \sqrt{|\lambda_{max}(A^T \cdot A)|}$ with: $\lambda_{max}$ is the largest Eigenvalue.
2. Abstraction (Continuation)

- We compute the spectral norm:
  - $||A|| = ||TAT^{-1}|| = ||B||$.
  - $B^T \cdot B = \begin{pmatrix} 10 \\ 11 \end{pmatrix} \begin{pmatrix} 11 \\ 01 \end{pmatrix} = \begin{pmatrix} 11 \\ 12 \end{pmatrix}$.
  - $\Rightarrow (2 - \lambda)(1 - \lambda) - 1 = 0$
  - $\Rightarrow \lambda^2 - 3\lambda + 1 = 0$
  - $\Rightarrow \lambda_{\text{max}}(B^T B) = \frac{3}{2} + \sqrt{\frac{5}{4}}$
  - $||A|| = \sqrt{\lambda_{\text{max}}(A^T A)} = \frac{1}{2}(1 + \sqrt{5})$
2. Abstraction (Continuation)

- We compute the spectral norm:
  \[ \|A\| = \|TAT^{-1}\| = \|B\| . \]
  
  - \[ B^T \cdot B = \begin{pmatrix} 10 \\ 11 \end{pmatrix} \begin{pmatrix} 11 \\ 01 \end{pmatrix} = \begin{pmatrix} 11 \\ 12 \end{pmatrix} . \]
  
  \[ \Rightarrow (2 - \lambda)(1 - \lambda) - 1 = 0 \]
  
  \[ \Rightarrow \lambda^2 - 3\lambda + 1 = 0 \]
  
  \[ \Rightarrow \lambda_{max}(B^T B) = \frac{3}{2} + \sqrt{\frac{5}{4}} \]
  
  \[ \|A\| = \sqrt{\lambda_{max}(A^T A)} = \frac{1}{2}(1 + \sqrt{5}) \]
2. Abstraction (Continuation)

- We compute the spectral norm:
  \[ \|A\| = \|TAT^{-1}\| = \|B\|. \]
- \[ B^\top \cdot B = \begin{pmatrix} 10 & 11 \\ 11 & 01 \end{pmatrix} = \begin{pmatrix} 11 & 12 \end{pmatrix}. \]
- \[ (2 - \lambda)(1 - \lambda) - 1 = 0 \]
- \[ \lambda^2 - 3\lambda + 1 = 0 \]
- \[ \Rightarrow \lambda_{\text{max}}(B^\top B) = \frac{3}{2} + \sqrt{\frac{5}{4}} \]
- \[ \|A\| = \sqrt{\lambda_{\text{max}}(A^\top A)} = \frac{1}{2}(1 + \sqrt{5}) \]
We compute the spectral norm:

- \(||A|| = ||TAT^{-1}|| = ||B||.

- \(B^T \cdot B = \begin{pmatrix} 10 \\ 11 \end{pmatrix} \begin{pmatrix} 11 \\ 01 \end{pmatrix} = \begin{pmatrix} 11 \\ 12 \end{pmatrix}\).

- \((2 - \lambda)(1 - \lambda) - 1 = 0\)

- \(\lambda^2 - 3\lambda + 1 = 0\)

- \(\lambda_{max}(B^T B) = \frac{3}{2} + \sqrt{\frac{5}{4}}\)

- \(||A|| = \sqrt{\lambda_{max}(A^T A)} = \frac{1}{2}(1 + \sqrt{5})\)
2. Abstraction (Continuation)

- We compute the spectral norm:
  - $\|A\| = \|TAT^{-1}\| = \|B\|$.
  - $B^T \cdot B = \begin{pmatrix} 10 \\ 11 \end{pmatrix} \begin{pmatrix} 11 \\ 01 \end{pmatrix} = \begin{pmatrix} 11 \\ 12 \end{pmatrix}$.
- $\Rightarrow (2 - \lambda)(1 - \lambda) - 1 = 0$
- $\Rightarrow \lambda^2 - 3\lambda + 1 = 0$
- $\Rightarrow \lambda_{max}(B^T B) = \frac{3}{2} + \sqrt{\frac{5}{4}}$
- $\|A\| = \sqrt{\lambda_{max}(A^T A)} = \frac{1}{2}(1 + \sqrt{5})$
2. Abstraction (Continuation)

- We compute the spectral norm:
  \[ ||A|| = ||TAT^{-1}|| = ||B||. \]
- \[ B^T \cdot B = \begin{pmatrix} 10 \\ 11 \end{pmatrix} \begin{pmatrix} 11 \\ 01 \end{pmatrix} = \begin{pmatrix} 11 \\ 12 \end{pmatrix}. \]
- \[ (2 - \lambda)(1 - \lambda) - 1 = 0 \]
- \[ \lambda^2 - 3\lambda + 1 = 0 \]
- \[ \Rightarrow \lambda_{max}(B^T B) = \frac{3}{2} + \sqrt{\frac{5}{4}} \]
- \[ ||A|| = \sqrt{\lambda_{max}(A^TA)} = \frac{1}{2}(1 + \sqrt{5}) \]
2. Abstraction (Continuation)

- We compute the spectral norm:
- $\| A \| = \| T A T^{-1} \| = \| B \|$.  
- $B^T \cdot B = \begin{pmatrix} 10 & 11 \\ 11 & 01 \end{pmatrix} = \begin{pmatrix} 11 \\ 12 \end{pmatrix}$. 
- $\Rightarrow (2 - \lambda)(1 - \lambda) - 1 = 0$ 
- $\Rightarrow \lambda^2 - 3\lambda + 1 = 0$ 
- $\Rightarrow \lambda_{\text{max}}(B^T B) = \frac{3}{2} + \sqrt{\frac{5}{4}}$ 
- $\| A \| = \sqrt{\lambda_{\text{max}}(A^T A)} = \frac{1}{2}(1 + \sqrt{5})$
Theorem:
A algorithm, solving the network counting problem needs \(2 + \lceil \log_2 \left( \frac{1 + \sqrt{5}}{2} \right)^n \rceil\) rounds.

Proof:
- Let \(A_j, 1 \leq j \leq r\) be matrices, which solve the problem in \(r\) rounds.
- \(\alpha := (\alpha_1, \alpha_2, \ldots, \alpha_n)^T = A_{r-2} \cdot \ldots \cdot A_2 \cdot A_1 \cdot (1, 1, \ldots, 1)\).
- \(\|\alpha\| \leq \left( \prod_{i=1}^{r-2} \|A_i\| \right) \cdot \|(1, \ldots, 1)\| \leq \left( \frac{1}{2} (1 + \sqrt{5}) \right)^{r-2} \cdot \sqrt{n}\)
- Let \(\inf(i, t)\) be the number, which have the nodes \(v_i\) after \(t\) rounds.
- After round \(t\) we have: \(\inf(i, t) \geq n\) for all \(i \in \{1, 2, \ldots, n\}\).
- After round \(t - 1\) we have: \(\inf(i, t - 1) \geq n\) for at least \(n/2\) nodes.
- There could be some \(i\) with: \(\inf(i, t - 2) \geq n\).
- But if \(\alpha_i < n\) and \(\inf(i, t - 1) \geq n\), then there exists \(j\) with: \(\alpha_i + \alpha_j \geq n\).
**Theorem:**

A algorithm, solving the network counting problem needs $2 + \left\lceil \log_{\frac{1}{2}}\left(1 + \sqrt{5}\right) \frac{n}{2} \right\rceil$ rounds.

**Proof:**

- Let $A_j, 1 \leq j \leq r$ be matrices, which solve the problem in $r$ rounds.
- $\alpha := (\alpha_1, \alpha_2, \cdots, \alpha_n)^T = A_{r-2} \cdot \ldots \cdot A_2 \cdot A_1 \cdot (1, 1, \cdots, 1)$.
- $||\alpha|| \leq (\prod_{i=1}^{r-2} ||A_i||) \cdot ||(1, \ldots, 1)|| \leq (\frac{1}{2}(1 + \sqrt{5}))^{r-2} \cdot \sqrt{n}$
- Let $\inf(i, t)$ be the number, which have the nodes $v_i$ after $t$ rounds.
- After round $t$ we have: $\inf(i, t) \geq n$ for all $i \in \{1, 2, \cdots, n\}$.
- After round $t - 1$ we have: $\inf(i, t - 1) \geq n$ for at least $n/2$ nodes.
- There could be some $i$ with: $\inf(i, t - 2) \geq n$.
- But if $\alpha_i < n$ and $\inf(i, t - 1) \geq n$, then there exists $j$ with: $\alpha_i + \alpha_j \geq n$. 
Theorem:

A algorithm, solving the network counting problem needs $2 + \left\lceil \log_{\frac{1}{2}}(1 + \sqrt{5}) \frac{n}{2} \right\rceil$ rounds.

Proof:

- Let $A_j, 1 \leq j \leq r$ be matrices, which solve the problem in $r$ rounds.
- $\alpha := (\alpha_1, \alpha_2, \cdots, \alpha_n)^T = A_{r-2} \cdot \cdots \cdot A_2 \cdot A_1 \cdot (1, 1, \cdots, 1)$.
- $||\alpha|| \leq \left( \prod_{i=1}^{r-2} ||A_i|| \right) \cdot ||(1, \ldots, 1)|| \leq \left( \frac{1}{2} (1 + \sqrt{5}) \right)^{r-2} \cdot \sqrt{n}$
- Let $\inf(i, t)$ be the number, which have the nodes $v_i$ after $t$ rounds.
- After round $t$ we have: $\inf(i, t) \geq n$ for all $i \in \{1, 2, \cdots, n\}$.
- After round $t - 1$ we have: $\inf(i, t - 1) \geq n$ for at least $n/2$ nodes.
- There could be some $i$ with: $\inf(i, t - 2) \geq n$.
- But if $\alpha_i < n$ and $\inf(i, t - 1) \geq n$, then there exists $j$ with: $\alpha_i + \alpha_j \geq n$. 
Theorem:
A algorithm, solving the network counting problem needs $2 + \lceil \log_{\frac{1}{2}}(1+\sqrt{5}) \frac{n}{2} \rceil$ rounds.

Proof:

- Let $A_j, 1 \leq j \leq r$ be matrices, which solve the problem in $r$ rounds.
- $\alpha := (\alpha_1, \alpha_2, \cdots, \alpha_n)^T = A_{r-2} \cdots A_2 \cdot A_1 \cdot (1, 1, \cdots, 1)$.
- $||\alpha|| \leq (\prod_{i=1}^{r-2} ||A_i||) \cdot ||(1, \ldots, 1)|| \leq (\frac{1}{2}(1 + \sqrt{5}))^{r-2} \cdot \sqrt{n}$
- Let $inf(i, t)$ be the number, which have the nodes $v_i$ after $t$ rounds.
- After round $t$ we have: $inf(i, t) \geq n$ for all $i \in \{1, 2, \cdots, n\}$.
- After round $t-1$ we have: $inf(i, t-1) \geq n$ for at least $n/2$ nodes.
- There could be some $i$ with: $inf(i, t-2) \geq n$.
- But if $\alpha_i < n$ and $inf(i, t-1) \geq n$, then there exists $j$ with: $\alpha_i + \alpha_j \geq n$. 

Theorem:

A algorithm, solving the network counting problem needs $2 + \lceil \log_{\frac{1}{2}}(1 + \sqrt{5}) \frac{n}{2} \rceil$ rounds.

Proof:

- Let $A_j, 1 \leq j \leq r$ be matrices, which solve the problem in $r$ rounds.
- $\alpha := (\alpha_1, \alpha_2, \cdots, \alpha_n)^T = A_{r-2} \cdots A_2 \cdot A_1 \cdot (1, 1, \cdots, 1)$.
- $||\alpha|| \leq (\prod_{i=1}^{r-2} ||A_i||) \cdot ||(1, \ldots, 1)|| \leq (\frac{1}{2}(1 + \sqrt{5}))^{r-2} \cdot \sqrt{n}$
- Let $\inf(i, t)$ be the number, which have the nodes $v_i$ after $t$ rounds.
- After round $t$ we have: $\inf(i, t) \geq n$ for all $i \in \{1, 2, \cdots, n\}$.
- After round $t - 1$ we have: $\inf(i, t - 1) \geq n$ for at least $n/2$ nodes.
- There could be some $i$ with: $\inf(i, t - 2) \geq n$.
- But if $\alpha_i < n$ and $\inf(i, t - 1) \geq n$, then there exists $j$ with: $\alpha_i + \alpha_j \geq n$. 
Theorem:

A algorithm, solving the network counting problem needs $2 + \lceil \log \frac{1}{2} (1 + \sqrt{5}) \frac{n}{2} \rceil$ rounds.

Proof:

- Let $A_j, 1 \leq j \leq r$ be matrices, which solve the problem in $r$ rounds.
- $\alpha := (\alpha_1, \alpha_2, \cdots, \alpha_n)^T = A_{r-2} \cdot \cdots \cdot A_2 \cdot A_1 \cdot (1, 1, \cdots, 1)$.
- $||\alpha|| \leq (\prod_{i=1}^{r-2} ||A_i||) \cdot ||(1, \cdots, 1)|| \leq \left( \frac{1}{2} (1 + \sqrt{5}) \right)^{r-2} \cdot \sqrt{n}$
- Let $\inf(i, t)$ be the number, which have the nodes $v_i$ after $t$ rounds.
- After round $t$ we have: $\inf(i, t) \geq n$ for all $i \in \{1, 2, \cdots, n\}$.
- After round $t - 1$ we have: $\inf(i, t - 1) \geq n$ for at least $n/2$ nodes.
- There could be some $i$ with: $\inf(i, t - 2) \geq n$.
- But if $\alpha_i < n$ and $\inf(i, t - 1) \geq n$, then there exists $j$ with: $\alpha_i + \alpha_j \geq n$. 
**Theorem:**

A algorithm, solving the network counting problem needs $2 + \lceil \log_{\frac{1}{2}} \left(1 + \sqrt{5}\right) \frac{n}{2} \rceil$ rounds.

**Proof:**

- Let $A_j, 1 \leq j \leq r$ be matrices, which solve the problem in $r$ rounds.
- $\alpha := (\alpha_1, \alpha_2, \cdots, \alpha_n)^T = A_{r-2} \cdot \cdots \cdot A_2 \cdot A_1 \cdot (1, 1, \cdots, 1)$.
- $||\alpha|| \leq \left( \prod_{i=1}^{r-2} ||A_i|| \right) \cdot ||(1, ..., 1)|| \leq \left( \frac{1}{2} (1 + \sqrt{5}) \right)^{r-2} \cdot \sqrt{n}$
- Let $\inf(i, t)$ be the number, which have the nodes $v_i$ after $t$ rounds.
- After round $t$ we have: $\inf(i, t) \geq n$ for all $i \in \{1, 2, \cdots, n\}$.
- After round $t - 1$ we have: $\inf(i, t - 1) \geq n$ for at least $n/2$ nodes.
- There could be some $i$ with: $\inf(i, t - 2) \geq n$.
- But if $\alpha_i < n$ and $\inf(i, t - 1) \geq n$, then there exists $j$ with: $\alpha_i + \alpha_j \geq n$. 


Theorem:

A algorithm, solving the network counting problem needs \(2 + \lceil \log_{\frac{1}{2}}(1+\sqrt{5}) \frac{n}{2} \rceil\) rounds.

Proof:

- Let \(A_j, 1 \leq j \leq r\) be matrices, which solve the problem in \(r\) rounds.
- \(\alpha := (\alpha_1, \alpha_2, \cdots, \alpha_n)^T = A_{r-2} \cdot \cdots \cdot A_2 \cdot A_1 \cdot (1, 1, \cdots, 1)\).
- \(\|\alpha\| \leq \left( \prod_{i=1}^{r-2} \|A_i\| \right) \cdot \|(1, \ldots, 1)\| \leq \left(\frac{1}{2}(1+\sqrt{5})\right)^{r-2} \cdot \sqrt{n}\)
- Let \(\inf(i, t)\) be the number, which have the nodes \(v_i\) after \(t\) rounds.
- After round \(t\) we have: \(\inf(i, t) \geq n\) for all \(i \in \{1, 2, \cdots, n\}\).
- After round \(t - 1\) we have: \(\inf(i, t - 1) \geq n\) for at least \(n/2\) nodes.
- There could be some \(i\) with: \(\inf(i, t - 2) \geq n\).
- But if \(\alpha_i < n\) and \(\inf(i, t - 1) \geq n\), then there exists \(j\) with: \(\alpha_i + \alpha_j \geq n\).
Theorem:

A algorithm, solving the network counting problem needs \(2 + \lceil \log_{\frac{1}{2}}(1+\sqrt{5}) \frac{n}{2} \rceil\) rounds.

Proof:

- Let \(A_j, 1 \leq j \leq r\) be matrices, which solve the problem in \(r\) rounds.
- \(\alpha := (\alpha_1, \alpha_2, \cdots, \alpha_n)^T = A_{r-2} \cdots A_2 \cdot A_1 \cdot (1, 1, \cdots, 1)\).
- \(||\alpha|| \leq (\prod_{i=1}^{r-2} ||A_i||) \cdot ||(1,\ldots,1)|| \leq \left(\frac{1}{2}(1+\sqrt{5})\right)^{r-2} \cdot \sqrt{n}\)
- Let \(\inf(i,t)\) be the number, which have the nodes \(v_i\) after \(t\) rounds.
- After round \(t\) we have: \(\inf(i,t) \geq n\) for all \(i \in \{1, 2, \cdots, n\}\).
- After round \(t-1\) we have: \(\inf(i,t-1) \geq n\) for at least \(n/2\) nodes.
- There could be some \(i\) with: \(\inf(i,t-2) \geq n\).
- But if \(\alpha_i < n\) and \(\inf(i,t-1) \geq n\), then there exists \(j\) with: \(\alpha_i + \alpha_j \geq n\).
Continuation

\[ \alpha := (\alpha_1, \alpha_2, \cdots, \alpha_n)^T = A_{r-2} \cdots A_2 \cdot A_1 \cdot (1, 1, \cdots, 1) \]

Let

- \( c_1 \) be the number of cases with: \( \alpha_i \geq n \),
- \( c_2 \) be the number of cases with: \( \alpha_i < n \) and \( \alpha_j \geq n \),
- \( c_3 \) be the number of cases with: \( \alpha_i < n \), \( \alpha_j < n \) and \( \alpha_i + \alpha_j \geq n \).

Then we have: \( c_1 \geq c_2 \) and \( c_1 + c_2 + c_3 \geq n/2 \).

Thus we also get: \( 2c_1 + c_3 \geq \frac{n}{2} \)

\[ ||\alpha|| = \sqrt{\sum_{i=1}^{n} \alpha_i^2} \geq \sqrt{c_1 n^2 + c_3 \cdot 2 \cdot \frac{n^2}{4}} \geq n \cdot \sqrt{\frac{1}{2}(2c_1 + c_3)} \geq \frac{n}{2} \sqrt{n}. \]

We already have:

\[ ||\alpha|| \leq \left( \prod_{i=1}^{r-2} ||A_i|| \right) \cdot ||(1, \ldots, 1)|| \leq \left( \frac{1}{2}(1 + \sqrt{5}) \right)^{r-2} \cdot \sqrt{n}. \]

And we get:

\[ \frac{n}{2} \cdot \sqrt{n} \leq ||\alpha|| \leq \Phi^{r-2} \cdot \sqrt{n}, \]

From which we conclude:

\[ r \geq 2 + \left[ \log_{{\frac{1}{2}}(1+\sqrt{5})} \frac{n}{2} \right] \]
Continuation

\[ \alpha := (\alpha_1, \alpha_2, \ldots, \alpha_n)^T = A_{r-2} \cdots A_2 \cdot A_1 \cdot (1, 1, \ldots, 1) \]

Let

- \( c_1 \) be the number of cases with: \( \alpha_i \geq n \),
- \( c_2 \) be the number of cases with: \( \alpha_i < n \) and \( \alpha_j \geq n \),
- \( c_3 \) be the number of cases with: \( \alpha_i < n, \alpha_j < n \) and \( \alpha_i + \alpha_j \geq n \).

Then we have: \( c_1 \geq c_2 \) and \( c_1 + c_2 + c_3 \geq n/2 \).

Thus we also get: \( 2c_1 + c_3 \geq \frac{n}{2} \)

\[ \|\alpha\| = \sqrt{\sum_{i=1}^{n} \alpha_i^2} \geq \sqrt{c_1 n^2 + c_3 \cdot 2 \cdot \frac{n^2}{4}} \geq n \cdot \sqrt{\frac{1}{2} (2c_1 + c_3)} \geq \frac{n}{2} \sqrt{n} \]

We already have:

\[ \|\alpha\| \leq (\prod_{i=1}^{r-2} \|A_i\|) \cdot \|(1, \ldots, 1)\| \leq (\frac{1}{2} (1 + \sqrt{5}))^{r-2} \cdot \sqrt{n}. \]

And we get:

\[ \frac{n}{2} \cdot \sqrt{n} \leq \|\alpha\| \leq \Phi^{r-2} \cdot \sqrt{n}, \]

From which we conclude:

\[ r \geq 2 + \left[ \log_{\frac{1}{2} (1 + \sqrt{5})} \frac{n}{2} \right] \]
Continuation

\[ \alpha := (\alpha_1, \alpha_2, \cdots, \alpha_n)^T = A_{r-2} \cdots A_2 \cdot A_1 \cdot (1, 1, \cdots, 1) \]

- Let
  - \( c_1 \) be the number of cases with: \( \alpha_i \geq n \),
  - \( c_2 \) be the number of cases with: \( \alpha_i < n \) and \( \alpha_j \geq n \),
  - \( c_3 \) be the number of cases with: \( \alpha_i < n, \alpha_j < n \) and \( \alpha_i + \alpha_j \geq n \).

- Then we have: \( c_1 \geq c_2 \) and \( c_1 + c_2 + c_3 \geq n/2 \).
- Thus we also get: \( 2c_1 + c_3 \geq \frac{n}{2} \)
- \[ ||\alpha|| = \sqrt{\sum_{i=1}^{n} \alpha_i^2} \geq \sqrt{c_1 n^2 + c_3 \cdot 2 \cdot \frac{n^2}{4}} \geq n \cdot \sqrt{\frac{1}{2}(2c_1 + c_3)} \geq \frac{n}{2} \sqrt{n}. \]
- We already have:
  \[ ||\alpha|| \leq \left( \prod_{i=1}^{r-2} ||A_i|| \right) \cdot ||(1, \ldots, 1)|| \leq \left( \frac{1}{2}(1 + \sqrt{5}) \right)^{r-2} \cdot \sqrt{n}. \]
- And we get:
  \[ \frac{n}{2} \cdot \sqrt{n} \leq ||\alpha|| \leq \Phi^{r-2} \cdot \sqrt{n}, \]
- From which we conclude:
  \[ r \geq 2 + \left\lceil \log_{\frac{1}{2}(1+\sqrt{5})} \frac{n}{2} \right\rceil \]
Let

- $c_1$ be the number of cases with: $\alpha_i \geq n$,
- $c_2$ be the number of cases with: $\alpha_i < n$ and $\alpha_j \geq n$,
- $c_3$ be the number of cases with: $\alpha_i < n$, $\alpha_j < n$ and $\alpha_i + \alpha_j \geq n$.

Then we have: $c_1 \geq c_2$ and $c_1 + c_2 + c_3 \geq n/2$.

Thus we also get: $2c_1 + c_3 \geq \frac{n}{2}$

$$||\alpha|| = \sqrt{\sum_{i=1}^{n} \alpha_i^2} \geq \sqrt{c_1 n^2 + c_3 \cdot 2 \cdot \frac{n^2}{4}} \geq n \cdot \sqrt{\frac{1}{2}(2c_1 + c_3)} \geq \frac{n}{2} \sqrt{n}.$$

We already have:

$$||\alpha|| \leq \prod_{i=1}^{r-2} ||A_i|| \cdot ||(1,\ldots,1)|| \leq \left(\frac{1}{2}(1 + \sqrt{5})\right)^{r-2} \cdot \sqrt{n}.$$

And we get:

$$\frac{n}{2} \cdot \sqrt{n} \leq ||\alpha|| \leq \Phi^{r-2} \cdot \sqrt{n},$$

From which we conclude:

$$r \geq 2 + \left\lceil \log \frac{1}{2}(1 + \sqrt{5}) \frac{n}{2} \right\rceil$$
\[ \alpha := (\alpha_1, \alpha_2, \ldots, \alpha_n)^T = A_{r-2} \cdots A_2 A_1 (1, 1, \ldots, 1) \]

- Let
  - \( c_1 \) be the number of cases with: \( \alpha_i \geq n \),
  - \( c_2 \) be the number of cases with: \( \alpha_i < n \) and \( \alpha_j \geq n \),
  - \( c_3 \) be the number of cases with: \( \alpha_i < n, \alpha_j < n \) and \( \alpha_i + \alpha_j \geq n \).

- Then we have: \( c_1 \geq c_2 \) and \( c_1 + c_2 + c_3 \geq n/2 \).

- Thus we also get: \( 2c_1 + c_3 \geq \frac{n}{2} \)

- \[ \|\alpha\| = \sqrt{\sum_{i=1}^{n} \alpha_i^2} \geq \sqrt{c_1 n^2 + c_3 \cdot 2 \cdot \frac{n^2}{4}} \geq n \cdot \sqrt{\frac{1}{2} (2c_1 + c_3)} \geq \frac{n}{2} \sqrt{n}. \]

- We already have:
  \[ \|\alpha\| \leq \left( \prod_{i=1}^{r-2} ||A_i|| \right) \cdot ||(1, \ldots, 1)|| \leq \left( \frac{1}{2} (1 + \sqrt{5}) \right)^{r-2} \cdot \sqrt{n}. \]

- And we get:
  \[ \frac{n}{2} \cdot \sqrt{n} \leq \|\alpha\| \leq \Phi^{r-2} \cdot \sqrt{n}, \]

- From which we conclude:
  \[ r \geq 2 + \left\lceil \log_{\frac{1}{2} (1 + \sqrt{5})} \frac{n}{2} \right\rceil \]
Continuation

\[ \alpha := (\alpha_1, \alpha_2, \ldots, \alpha_n)^T = A_{r-2} \cdots A_2 \cdot A_1 \cdot (1, 1, \ldots, 1) \]

- Let
  - \( c_1 \) be the number of cases with: \( \alpha_i \geq n \),
  - \( c_2 \) be the number of cases with: \( \alpha_i < n \) and \( \alpha_j \geq n \),
  - \( c_3 \) be the number of cases with: \( \alpha_i < n \), \( \alpha_j < n \) and \( \alpha_i + \alpha_j \geq n \).

Then we have: \( c_1 \geq c_2 \) and \( c_1 + c_2 + c_3 \geq n/2 \).

Thus we also get: \( 2c_1 + c_3 \geq \frac{n}{2} \).

- \[ ||\alpha|| = \sqrt{\sum_{i=1}^{n} \alpha_i^2} \geq \sqrt{c_1 n^2 + c_3 \cdot 2 \cdot \frac{n^2}{4}} \geq n \cdot \sqrt{\frac{1}{2}(2c_1 + c_3)} \geq \frac{n}{2} \sqrt{n}. \]

- We already have:
  \[ ||\alpha|| \leq \left( \prod_{i=1}^{r-2} ||A_i|| \right) \cdot ||(1, \ldots, 1)|| \leq \left( \frac{1}{2}(1 + \sqrt{5}) \right)^{r-2} \cdot \sqrt{n}. \]

- And we get:
  \[ \frac{n}{2} \cdot \sqrt{n} \leq ||\alpha|| \leq \Phi^{r-2} \cdot \sqrt{n}, \]

- From which we conclude:
  \[ r \geq 2 + \lceil \log_{\frac{1}{2}(1+\sqrt{5})} \frac{n}{2} \rceil \]
Continuation

\[ \alpha := (\alpha_1, \alpha_2, \cdots, \alpha_n)^T = A_{r-2} \cdots A_2 A_1 \cdot (1, 1, \cdots, 1) \]

- Let
  - \( c_1 \) be the number of cases with: \( \alpha_i \geq n \),
  - \( c_2 \) be the number of cases with: \( \alpha_i < n \) and \( \alpha_j \geq n \),
  - \( c_3 \) be the number of cases with: \( \alpha_i < n, \alpha_j < n \) and \( \alpha_i + \alpha_j \geq n \).

- Then we have: \( c_1 \geq c_2 \) and \( c_1 + c_2 + c_3 \geq \frac{n}{2} \).
- Thus we also get: \( 2c_1 + c_3 \geq \frac{n}{2} \).

- \[ \|\alpha\| = \sqrt{\sum_{i=1}^{n} \alpha_i^2} \geq \sqrt{c_1 n^2 + c_3 \cdot 2 \cdot \frac{n^2}{4}} \geq n \cdot \sqrt{\frac{1}{2}(2c_1 + c_3)} \geq \frac{n}{2} \sqrt{n}. \]
- We already have:
  \[ \|\alpha\| \leq (\prod_{i=1}^{r-2} \|A_i\|) \cdot \|(1, \ldots, 1)\| \leq (\frac{1}{2}(1 + \sqrt{5}))^{r-2} \cdot \sqrt{n}. \]
- And we get:
  \[ \frac{n}{2} \cdot \sqrt{n} \leq \|\alpha\| \leq \Phi^{r-2} \cdot \sqrt{n}, \]
- From which we conclude:
  \[ r \geq 2 + \left\lceil \log_{\frac{1}{2}(1+\sqrt{5})} \frac{n}{2} \right\rceil \]
Let
- $c_1$ be the number of cases with: $\alpha_i \geq n$,
- $c_2$ be the number of cases with: $\alpha_i < n$ and $\alpha_j \geq n$,
- $c_3$ be the number of cases with: $\alpha_i < n$, $\alpha_j < n$ and $\alpha_i + \alpha_j \geq n$.

Then we have: $c_1 \geq c_2$ and $c_1 + c_2 + c_3 \geq n/2$.

Thus we also get: $2c_1 + c_3 \geq \frac{n}{2}$

$$||\alpha|| = \sqrt{\sum_{i=1}^{n} \alpha_i^2} \geq \sqrt{c_1 n^2 + c_3 \cdot 2 \cdot \frac{n^2}{4}} \geq n \cdot \sqrt{\frac{1}{2} (2c_1 + c_3)} \geq \frac{n \sqrt{n}}{2}.$$  

We already have:

$$||\alpha|| \leq \left( \prod_{i=1}^{r-2} ||A_i|| \right) \cdot ||(1, ..., 1)|| \leq \left( \frac{1}{2} (1 + \sqrt{5}) \right)^{r-2} \cdot \sqrt{n}.$$  

And we get:

$$\frac{n}{2} \cdot \sqrt{n} \leq ||\alpha|| \leq \Phi^{r-2} \cdot \sqrt{n},$$

From which we conclude:

$$r \geq 2 + \left[ \log \frac{1}{2} (1 + \sqrt{5}) \frac{n}{2} \right]$$
Continuation

\[ \alpha := (\alpha_1, \alpha_2, \cdots, \alpha_n)^T = A_{r-2} \cdots A_2 A_1 \cdot (1, 1, \cdots, 1) \]

- Let
  - \( c_1 \) be the number of cases with: \( \alpha_i \geq n \),
  - \( c_2 \) be the number of cases with: \( \alpha_i < n \) and \( \alpha_j \geq n \),
  - \( c_3 \) be the number of cases with: \( \alpha_i < n, \alpha_j < n \) and \( \alpha_i + \alpha_j \geq n \).

Then we have: \( c_1 \geq c_2 \) and \( c_1 + c_2 + c_3 \geq n/2 \).

Thus we also get: \( 2c_1 + c_3 \geq \frac{n}{2} \)

- \[ ||\alpha|| = \sqrt{\sum_{i=1}^{n} \alpha_i^2} \geq \sqrt{c_1 n^2 + c_3 \cdot 2 \cdot \frac{n^2}{4}} \geq n \cdot \sqrt{\frac{1}{2} (2c_1 + c_3)} \geq \frac{n}{2} \sqrt{n}. \]

We already have:

- \[ ||\alpha|| \leq \left( \prod_{i=1}^{r-2} ||A_i|| \right) \cdot ||(1, \ldots, 1)|| \leq \left( \frac{1}{2} (1 + \sqrt{5}) \right)^{r-2} \cdot \sqrt{n}. \]

And we get:

- \[ \frac{n}{2} \cdot \sqrt{n} \leq ||\alpha|| \leq \Phi^{r-2} \cdot \sqrt{n}, \]

From which we conclude:

\[ r \geq 2 + \left[ \log_{\frac{1}{2} (1 + \sqrt{5})} \frac{n}{2} \right] \]
\[\alpha := (\alpha_1, \alpha_2, \ldots, \alpha_n)^T = A_{r-2} \cdots A_2 \cdot A_1 \cdot (1, 1, \ldots, 1)\]

- Let
  - \(c_1\) be the number of cases with: \(\alpha_i \geq n\),
  - \(c_2\) be the number of cases with: \(\alpha_i < n\) and \(\alpha_j \geq n\),
  - \(c_3\) be the number of cases with: \(\alpha_i < n\), \(\alpha_j < n\) and \(\alpha_i + \alpha_j \geq n\).

Then we have: \(c_1 \geq c_2\) and \(c_1 + c_2 + c_3 \geq \frac{n}{2}\).

Thus we also get: \(2c_1 + c_3 \geq \frac{n}{2}\)

\[\|\alpha\| = \sqrt{\sum_{i=1}^{n} \alpha_i^2} \geq \sqrt{c_1 n^2 + c_3 \cdot 2 \cdot \frac{n^2}{4}} \geq n \cdot \sqrt{\frac{1}{2}(2c_1 + c_3)} \geq \frac{n}{2} \cdot \sqrt{n}.\]

We already have:

\[\|\alpha\| \leq \left( \prod_{i=1}^{r-2} ||A_i|| \right) \cdot ||(1, \ldots, 1)|| \leq (\frac{1}{2}(1 + \sqrt{5}))^{r-2} \cdot \sqrt{n}.\]

And we get:

\[\frac{n}{2} \cdot \sqrt{n} \leq \|\alpha\| \leq \Phi^{r-2} \cdot \sqrt{n},\]

From which we conclude:

\[r \geq 2 + \left\lceil \log_{\frac{1}{2}(1+\sqrt{5})} \frac{n}{2} \right\rceil\]
Quality of these Bounds

Lemma:

Let \( n = 2m \) and let:

- \( t_1 := 1 + k \), with \( k \) is the smallest number with \( m \leq F(k) \) and
- \( t_2 := 2 + \lceil \log_{\frac{1}{2}}(1+\sqrt{5}) m \rceil \).

Then we have \( t_1 = t_2 \) for infinite many \( m \) and \( t_1 \leq t_2 + 1 \) for all \( m \).

Proof:

- Let \( \Phi = \frac{1}{2}(1+\sqrt{5}) \).
- Then we have: \( \Phi^2 = \Phi + 1 \).
- Furthermore we have \( \Phi^{i-2} \leq F(i) \leq \Phi^{i-1} \) for all \( i \geq 2 \).
- Consider \( n \in \mathbb{N} \) with: \( n = 2 \cdot F(k) \) for some \( k \).
  - Then we have: \( t_1 = k + 1 \) and
    \[
    t_2 = 2 + \lceil \log_{\Phi} F(k) \rceil = 2 + k - 1 = k + 1.
    \]
  - From which we get: \( t_1 = t_2 \) for these \( n \).
Quality of these Bounds

Lemma:
Let $n = 2m$ and let:
- $t_1 := 1 + k$, with $k$ is the smallest number with $m \leq F(k)$ and
- $t_2 := 2 + \lceil \log \frac{1}{2} (1 + \sqrt{5}) m \rceil$.

Then we have $t_1 = t_2$ for infinite many $m$ and $t_1 \leq t_2 + 1$ for all $m$.

Proof:
- Let $\Phi = \frac{1}{2} (1 + \sqrt{5})$.
- Then we have: $\Phi^2 = \Phi + 1$.
- Furthermore we have $\Phi^{i-2} \leq F(i) \leq \Phi^{i-1}$ for all $i \geq 2$.
- Consider $n \in \mathbb{N}$ with: $n = 2 \cdot F(k)$ for some $k$.
  - Then we have: $t_1 = k + 1$ and $t_2 = 2 + \lceil \log \Phi F(k) \rceil = 2 + k - 1 = k + 1$.
  - From which we get: $t_1 = t_2$ for these $n$. 

Quality of these Bounds

Lemma:

Let $n = 2m$ and let:

1. $t_1 := 1 + k$, with $k$ is the smallest number with $m \leq F(k)$ and
2. $t_2 := 2 + \lceil \log_{\frac{1}{2}}(1 + \sqrt{5}) m \rceil$.

Then we have $t_1 = t_2$ for infinite many $m$ and $t_1 \leq t_2 + 1$ for all $m$.

Proof:

1. Let $\Phi = \frac{1}{2} (1 + \sqrt{5})$.
2. Then we have: $\Phi^2 = \Phi + 1$.
3. Furthermore we have $\Phi^{i-2} \leq F(i) \leq \Phi^{i-1}$ for all $i \geq 2$.
4. Consider $n \in \mathbb{N}$ with: $n = 2 \cdot F(k)$ for some $k$.

   - Then we have: $t_1 = k + 1$ and
     
     $t_2 = 2 + \lceil \log_{\Phi} F(k) \rceil = 2 + k - 1 = k + 1$.

   - From which we get: $t_1 = t_2$ for these $n$. 
Quality of these Bounds

**Lemma:**

Let $n = 2m$ and let:

- $t_1 := 1 + k$, with $k$ is the smallest number with $m \leq F(k)$ and
- $t_2 := 2 + \lceil \log_{\frac{1}{2}}(1 + \sqrt{5}) m \rceil$.

Then we have $t_1 = t_2$ for infinite many $m$ and $t_1 \leq t_2 + 1$ for all $m$.

**Proof:**

- Let $\Phi = \frac{1}{2} (1 + \sqrt{5})$.
- Then we have: $\Phi^2 = \Phi + 1$.
- Furthermore we have $\Phi^{i-2} \leq F(i) \leq \Phi^{i-1}$ for all $i \geq 2$.
- Consider $n \in \mathbb{N}$ with: $n = 2 \cdot F(k)$ for some $k$.
  - Then we have: $t_1 = k + 1$ and $t_2 = 2 + \lceil \log_{\Phi} F(k) \rceil = 2 + k - 1 = k + 1$.
  - From which we get: $t_1 = t_2$ for these $n$. 
Quality of these Bounds

Lemma:

Let \( n = 2m \) and let:

\[ t_1 := 1 + k, \text{ with } k \text{ is the smallest number with } m \leq F(k) \text{ and} \]
\[ t_2 := 2 + \lceil \log_{\frac{1}{2}} \left( 1 + \sqrt{5} \right) m \rceil. \]

Then we have \( t_1 = t_2 \) for infinite many \( m \) and \( t_1 \leq t_2 + 1 \) for all \( m \).

Proof:

- Let \( \Phi = \frac{1}{2} (1 + \sqrt{5}) \).
- Then we have: \( \Phi^2 = \Phi + 1 \).
- Furthermore we have \( \Phi^{i-2} \leq F(i) \leq \Phi^{i-1} \) for all \( i \geq 2 \).
- Consider \( n \in \mathbb{N} \) with: \( n = 2 \cdot F(k) \) for some \( k \).
  - Then we have: \( t_1 = k + 1 \) and \( t_2 = 2 + \lceil \log_\Phi F(k) \rceil = 2 + k - 1 = k + 1 \).
  - From which we get: \( t_1 = t_2 \) for these \( n \).
Quality of these Bounds

Lemma:

Let $n = 2m$ and let:

1. $t_1 := 1 + k$, with $k$ is the smallest number with $m \leq F(k)$ and
2. $t_2 := 2 + \lceil \log_{\frac{1}{2}}(1 + \sqrt{5}) m \rceil$.

Then we have $t_1 = t_2$ for infinite many $m$ and $t_1 \leq t_2 + 1$ for all $m$.

Proof:

1. Let $\Phi = \frac{1}{2}(1 + \sqrt{5})$.
2. Then we have: $\Phi^2 = \Phi + 1$.
3. Furthermore we have $\Phi^{i-2} \leq F(i) \leq \Phi^{i-1}$ for all $i \geq 2$.
4. Consider $n \in \mathbb{N}$ with: $n = 2 \cdot F(k)$ for some $k$.
   1. Then we have: $t_1 = k + 1$ and $t_2 = 2 + \lceil \log_{\Phi} F(k) \rceil = 2 + k - 1 = k + 1$.
   2. From which we get: $t_1 = t_2$ for these $n$. 

Quality of these Bounds

Lemma:

Let \( n = 2m \) and let:

- \( t_1 := 1 + k \), with \( k \) is the smallest number with \( m \leq F(k) \) and
- \( t_2 := 2 + \lceil \log_{\frac{1}{2}}(1 + \sqrt{5}) \cdot m \rceil \).

Then we have \( t_1 = t_2 \) for infinite many \( m \) and \( t_1 \leq t_2 + 1 \) for all \( m \).

Proof:

- Let \( \Phi = \frac{1}{2}(1 + \sqrt{5}) \).
- Then we have: \( \Phi^2 = \Phi + 1 \).
- Furthermore we have \( \Phi^{i-2} \leq F(i) \leq \Phi^{i-1} \) for all \( i \geq 2 \).
- Consider \( n \in \mathbb{N} \) with: \( n = 2 \cdot F(k) \) for some \( k \).
  - Then we have: \( t_1 = k + 1 \) and \( t_2 = 2 + \lceil \log_{\Phi} F(k) \rceil = 2 + k - 1 = k + 1 \).
  - From which we get: \( t_1 = t_2 \) for these \( n \).
Quality of these Bounds

Lemma:

Let $n = 2m$ and let:

- $t_1 := 1 + k$, with $k$ is the smallest number with $m \leq F(k)$ and
- $t_2 := 2 + \lceil \log_{\frac{1}{2}}(1+\sqrt{5}) m \rceil$.

Then we have $t_1 = t_2$ for infinite many $m$ and $t_1 \leq t_2 + 1$ for all $m$.

Proof:

- Let $\Phi = \frac{1}{2}(1 + \sqrt{5})$.
- Then we have: $\Phi^2 = \Phi + 1$.
- Furthermore we have $\Phi^{i-2} \leq F(i) \leq \Phi^{i-1}$ for all $i \geq 2$.
- Consider $n \in \mathbb{N}$ with: $n = 2 \cdot F(k)$ for some $k$.
  - Then we have: $t_1 = k + 1$ and $t_2 = 2 + \lceil \log_{\Phi} F(k) \rceil = 2 + k - 1 = k + 1$.
  - From which we get: $t_1 = t_2$ for these $n$. 
Quality of these Bounds

**Lemma:**

Let $n = 2m$ and let:

- $t_1 := 1 + k$, with $k$ is the smallest number with $m \leq F(k)$ and
- $t_2 := 2 + \lceil \log_{\frac{1}{2}}(1 + \sqrt{5}) \cdot m \rceil$.

Then we have $t_1 = t_2$ for infinite many $m$ and $t_1 \leq t_2 + 1$ for all $m$.

**Proof:**

- Let $\Phi = \frac{1}{2}(1 + \sqrt{5})$.
- Then we have: $\Phi^2 = \Phi + 1$.
- Furthermore we have $\Phi^{i-2} \leq F(i) \leq \Phi^{i-1}$ for all $i \geq 2$.
- Consider $n \in \mathbb{N}$ with: $n = 2 \cdot F(k)$ for some $k$.
  - Then we have: $t_1 = k + 1$ and
    $t_2 = 2 + \lceil \log_\Phi F(k) \rceil = 2 + k - 1 = k + 1$.
  - From which we get: $t_1 = t_2$ for these $n$. 
Quality of these Bounds

Lemma:

Let \( n = 2m \) and let:

- \( t_1 := 1 + k \), with \( k \) is the smallest number with \( m \leq F(k) \) and
- \( t_2 := 2 + \lceil \log_{\frac{1}{2}}(1 + \sqrt{5}) m \rceil \).

Then we have \( t_1 = t_2 \) for infinite many \( m \) and \( t_1 \leq t_2 + 1 \) for all \( m \).

Proof:

- Let \( \Phi = \frac{1}{2}(1 + \sqrt{5}) \).
- Then we have: \( \Phi^2 = \Phi + 1 \).
- Furthermore we have \( \Phi^{i-2} \leq F(i) \leq \Phi^{i-1} \) for all \( i \geq 2 \).
- Consider \( n \in \mathbb{N} \) with: \( n = 2 \cdot F(k) \) for some \( k \).
  - Then we have: \( t_1 = k + 1 \) and 
    \[ t_2 = 2 + \lceil \log_{\Phi} F(k) \rceil = 2 + k - 1 = k + 1. \]
  - From which we get: \( t_1 = t_2 \) for these \( n \).
Quality of these Bounds (Part 2)

**Lemma:**

Let $n = 2m$ and let:

- $t_1 := 1 + k$, with $k$ is the smallest number with $m \leq F(k)$ and
- $t_2 := 2 + \lceil \log_{\frac{1}{2}}(1 + \sqrt{5}) m \rceil$.

Then we have $t_1 = t_2$ for infinite many $m$ and $t_1 \leq t_2 + 1$ for all $m$.

**Proof:**

- Setze $\Phi = \frac{1}{2}(1 + \sqrt{5})$.
- Then we have $\Phi^{i-2} \leq F(i) \leq \Phi^{i-1}$ for all $i \geq 2$.
- Let $n = 2 \cdot m$ arbitrary.
  - Let $i$ be defined by: $\Phi^{i-1} < m \leq \Phi^i$, then we have: $t_2 = 2 + i$.
  - Let $k$ be the smallest number with $F(k) \geq m$.
  - Note: $\Phi^{k-2} \leq F(k) \leq \Phi^{k-1}$.
  - Then we have: $i = k - 1$ oder $i = k - 2$.
  - From which we conclude: $t_1 = k + 1 \leq i + 3$. 
Lemma:
Let $n = 2m$ and let:
- $t_1 := 1 + k$, with $k$ is the smallest number with $m \leq F(k)$ and
- $t_2 := 2 + \lceil \log_{\frac{1}{2}}(1+\sqrt{5}) m \rceil$.

Then we have $t_1 = t_2$ for infinite many $m$ and $t_1 \leq t_2 + 1$ for all $m$.

Proof:
- Setze $\Phi = \frac{1}{2}(1 + \sqrt{5})$.
- Then we have $\Phi^{i-2} \leq F(i) \leq \Phi^{i-1}$ for all $i \geq 2$.
- Let $n = 2 \cdot m$ arbitrary.
  - Let $i$ be defined by: $\Phi^{i-1} < m \leq \Phi^i$, then we have: $t_2 = 2 + i$.
  - Let $k$ be the smallest number with $F(k) \geq m$.
  - Note: $\Phi^{k-2} \leq F(k) \leq \Phi^{k-1}$.
  - Then we have: $i = k - 1$ oder $i = k - 2$.
  - From which we conclude: $t_1 = k + 1 \leq i + 3$. 
Quality of these Bounds (Part 2)

Lemma:

Let $n = 2m$ and let:

- $t_1 := 1 + k$, with $k$ is the smallest number with $m \leq F(k)$ and
- $t_2 := 2 + \lceil \log_{\frac{1}{2}}(1+\sqrt{5}) m \rceil$.

Then we have $t_1 = t_2$ for infinite many $m$ and $t_1 \leq t_2 + 1$ for all $m$.

Proof:

- Setze $\Phi = \frac{1}{2}(1 + \sqrt{5})$.
- Then we have $\Phi^{i-2} \leq F(i) \leq \Phi^{i-1}$ for all $i \geq 2$.
- Let $n = 2 \cdot m$ arbitrary.
  - Let $i$ be defined by: $\Phi^{i-1} < m \leq \Phi^i$, then we have: $t_2 = 2 + i$.
  - Let $k$ be the smallest number with $F(k) \geq m$.
  - Note: $\Phi^{k-2} \leq F(k) \leq \Phi^{k-1}$.
  - Then we have: $i = k - 1$ oder $i = k - 2$.
  - From which we conclude: $t_1 = k + 1 \leq i + 3$. 
Lemma:

Let \( n = 2m \) and let:
- \( t_1 := 1 + k \), with \( k \) is the smallest number with \( m \leq F(k) \) and
- \( t_2 := 2 + \left\lceil \log_{\frac{1}{2}}(1 + \sqrt{5}) \right\rceil m \).

Then we have \( t_1 = t_2 \) for infinite many \( m \) and \( t_1 \leq t_2 + 1 \) for all \( m \).

Proof:
- Setze \( \Phi = \frac{1}{2}(1 + \sqrt{5}) \).
- Then we have \( \Phi^{i-2} \leq F(i) \leq \Phi^{i-1} \) for all \( i \geq 2 \).
- Let \( n = 2 \cdot m \) arbitrary.
  - Let \( i \) be defined by: \( \Phi^{i-1} < m \leq \Phi^i \), then we have: \( t_2 = 2 + i \).
  - Let \( k \) be the smallest number with \( F(k) \geq m \).
  - Note: \( \Phi^{k-2} \leq F(k) \leq \Phi^{k-1} \).
  - Then we have: \( i = k - 1 \) oder \( i = k - 2 \).
  - From which we conclude: \( t_1 = k + 1 \leq i + 3 \).
Quality of these Bounds (Part 2)

Lemma:

Let $n = 2m$ and let:

- $t_1 := 1 + k$, with $k$ is the smallest number with $m \leq F(k)$ and
- $t_2 := 2 + \lceil \log_{\frac{1}{2}}(1+\sqrt{5}) m \rceil$.

Then we have $t_1 = t_2$ for infinite many $m$ and $t_1 \leq t_2 + 1$ for all $m$.

Proof:

- Setze $\Phi = \frac{1}{2}(1 + \sqrt{5})$.
- Then we have $\Phi^{i-2} \leq F(i) \leq \Phi^{i-1}$ for all $i \geq 2$.
- Let $n = 2 \cdot m$ arbitrary.
  - Let $i$ be defined by: $\Phi^{i-1} < m \leq \Phi^i$, then we have: $t_2 = 2 + i$.
  - Let $k$ be the smallest number with $F(k) \geq m$.
  - Note: $\Phi^{k-2} \leq F(k) \leq \Phi^{k-1}$.
  - Then we have: $i = k - 1$ oder $i = k - 2$.
  - From which we conclude: $t_1 = k + 1 \leq i + 3$. 
## Summary (Telefon-Mode)

| Graph | $|V|$ | diam | Lower Bound | Upper Bound |
|-------|------|------|-------------|-------------|
| $K_n$ | $n$  | 1    | $\lceil \log_2 n \rceil + \text{odd}(n)$ | $\lceil \log_2 n \rceil + \text{odd}(n)$ |
| $H_k$ | $2^k$ | $k$  | $n - \text{even}(n)$ | $n - \text{even}(n)$ |
| $P_n$ | $n$  | $n - 1$ | $\left\lfloor \frac{n}{2} \right\rfloor + \text{odd}(n)$ | $\left\lfloor \frac{n}{2} \right\rfloor + \text{odd}(n)$ |
| $C_n$ | $n$  | $\left\lfloor \frac{n}{2} \right\rfloor - 2$ | $\left\lfloor \frac{5k}{2} \right\rfloor - 2, k \text{ even}$ |
| $CCC_k$ | $k \cdot 2^k$ | $\left\lfloor \frac{5k}{2} \right\rfloor + 1, k \text{ odd}$ | $2k + 5$ |
| $SE_k$ | $2^k$ | $2k - 1$ | $1.9770k$ | $2.25 \cdot k + o(k)$ |
| $BF_k$ | $k \cdot 2^k$ | $\left\lfloor \frac{3k}{2} \right\rfloor$ | $1.5965k$ | $2k + 5$ |
| $DB_k$ | $2^k$ | $k$ | $1.5965k$ | $2k + 5$ |
## Summary (Telegraph-Mode)

| Graph | $|V|$ | diam | Lower Bound | Upper Bound |
|-------|------|------|-------------|-------------|
| $K_n$ | $n$  | 1    | $1.44 \log_2 n$ | $1.44 \log_2 n$ |
| $H_k$ | $2^k$ | $k$  | $1.44k$     | $1.88k$     |
| $P_n$ | $n$  | $n-1$| $n + \text{odd}(n)$ | $n + \text{odd}(n)$ |
| $C_n$ | $n$ even | $\Ceil{\frac{n}{2}}$ | $\Ceil{\frac{n}{2}} + \Ceil{\sqrt{2n} - \frac{1}{2}} - 1$ | $\Ceil{\frac{n}{2}} + \Ceil{\sqrt{2n} - \frac{1}{2}} - 1$ |
|       | $n$ odd | $\Ceil{\frac{n}{2}}$ | $\Ceil{\frac{n}{2}} + \Ceil{\sqrt{2n} - \frac{1}{2}} - 1$ | $\Ceil{\frac{n}{2}} + \Ceil{\sqrt{2n} - \frac{1}{2}} - 1$ |
| $CCC_k$ | $k \cdot 2^k$ | $\Ceil{\frac{5k}{2}} - 2$ | $\Ceil{\frac{5k}{2}} - 2$ | $\Ceil{\frac{7k}{2}} + \Ceil{2\sqrt{\Ceil{\frac{k}{2}}} - 2}$ |
| $SE_k$  | $2^k$ | $2k-1$ | $2k-1$ | $3k + 3$ |
| $BF_k$  | $k \cdot 2^k$ | $\Ceil{\frac{3k}{2}}$ | $1.9770k$ | $\Ceil{\frac{5k}{2}} + \Ceil{2\sqrt{\Ceil{\frac{k}{2}}} - 1}$ |
| $DB_k$  | $2^k$ | $k$  | $1.5965k$ | $3k + 3$ |
Lemma

\[ edp-r_1(G) \leq \min_{u \in V(G)} \{ edp-a_u(G) + edp-b_u(G) \} \]
\[ = 2 \cdot edp-b_{\min}(G) \]
\[ edp-r_2(G) \leq 2 \cdot edp-b_{\min}(G) - 1 \]

Lemma

For any graph \( G_n \) of \( n \) nodes, \( n \geq 2 \),
- \( \lceil \log_2 n \rceil \leq edp-r_2(G_n) \leq 2 \cdot \lceil \log_2 n \rceil + 1 \),
- \( \log_b(\lfloor n/2 \rfloor) + 2 \leq edp-r_1(G_n) \leq 2 \cdot \lceil \log_2 n \rceil + 2 \).
Results

Lemma

For each complete binary tree $T^h_2$ of depth $h \geq 3$ (and $n = 2^{h+1} - 1$ nodes),

- $2h + 3 = 2 \cdot \lceil \log_2 n \rceil + 1 \leq edp_r_1(C2T_h) \leq 2h + 4$,
- $2h + 2 = 2 \cdot \lceil \log_2 n \rceil \leq edp_r_2(C2T_h) \leq 2h + 3$.

Lemma

\[
edp_r_2(Gr^n_2) = 1.5 \cdot \log_2 n - \log_2 \log_2 n \pm O(1)\]
\[
edp_r_2(Pl(n, h)) \geq 1.5 \log_2 n - \log_2 \log_2 n - 0.5 \log_2 h - 2\]
Results

Lemma

For $d \geq 3$

(i) $edp_{r_2}(Gr^n_d) = (1 + 1/d) \cdot \log_2 n - \log_2 n \log_2 n \pm O(d)$,

(ii) $edp_{r_1}(Gr^n_d) \leq (\log_b 2 + (2 - \log_b 2)/d) \cdot \log_2 n + O(d) = (1.44... + 0.56.../d) \cdot \log_2 n + O(d)$.

Lemma

For every $X_k \in \{BF_k, CCC_k, Q_k\}$ of $n$ nodes and dimension $k$,

$edp_{r_1}(X_k) \leq r_1(K_n) + O(\log_2 \log_2 n)$.

Lemma

For every $Y_k \in \{BF_k, CCC_k\}$ of $n$ nodes and dimension $k$,

$edp_{r_2}(Y_k) \leq r_2(K_n) + O(\log_2 \log_2 n)$.
Literature

Legend

- : Not of relevance
- : implicitly used basics
- : idea of proof or algorithm
- : structure of proof or algorithm
- : Full knowledge