Contents I

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

2. Broadcast
   - Lower Bound
   - First Results
   - Trees

3. Complexity

4. Broadcast on Networks
   - Definition and first Results
   - Complexity

5. Lower Bounds
   - Degree of the Nodes
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 $\bigcup_{w \in V} I(w)$.
### 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 $I(w)$
- 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 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 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 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 $\bigcup_{w \in V} I(w)$.

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 $\bigcup_{w \in V} I(w)$.

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 $\bigcup_{w \in V} I(w)$. 
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

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

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

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

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

- $b_2(v, G) = \min\{\text{comm}(A) | \ A \text{ is a two-way algorithm for the broadcast-problem on } G \text{ and } v\}$

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

- $a_2(v, G) = \min\{\text{comm}(A) | \ A \text{ is a two-way algorithm for the accumulations-problem on } G \text{ and } v\}$
Definition

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

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

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

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

- $b_2(v, G) = \min \{ \text{comm}(A) \mid A \text{ is a two-way algorithm for the broadcast-problem on } G \text{ and } v \}$

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

- $a_2(v, G) = \min \{ \text{comm}(A) \mid A \text{ is a two-way algorithm for the accumulations-problem on } G \text{ and } v \}$
By $\text{comm}(A)$ we denote the complexity (number of rounds) of a communication-algorithm.

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

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

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

$b_2(v, G) = \min \{ \text{comm}(A) \mid A \text{ is a two-way algorithm for the broadcast-problem on } G \text{ and } v \}$

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

$a_2(v, G) = \min \{ \text{comm}(A) \mid A \text{ is a two-way algorithm for the accumulations-problem on } G \text{ and } v \}$
Definition

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

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

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

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

- $b_2(v, G) = \min\{\text{comm}(A) \mid A \text{ is a two-way algorithm for the broadcast-problem on } G \text{ and } v\}$

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

- $a_2(v, G) = \min\{\text{comm}(A) \mid A \text{ is a two-way algorithm for the accumulations-problem on } G \text{ and } v\}$
Definition

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

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

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

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

- $b_2(v, G) = \min\{\text{comm}(A) \mid A \text{ is a two-way algorithm for the broadcast-problem on } G \text{ and } v\}$

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

- $a_2(v, G) = \min\{\text{comm}(A) \mid A \text{ is a two-way algorithm for the accumulations-problem on } G \text{ and } v\}$
By \( \text{comm}(A) \) we denote the complexity (number of rounds) of a communication-algorithm.

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

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

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

\[ b_2(v, G) = \min \{ \text{comm}(A) \mid A \text{ is a two-way algorithm for the broadcast-problem on } G \text{ and } v \} \]

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

\[ a_2(v, G) = \min \{ \text{comm}(A) \mid A \text{ is a two-way algorithm for the accumulations-problem on } G \text{ and } v \} \]
Definition

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

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

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

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

- $b_2(v, G) = \min\{\text{comm}(A) \mid A \text{ is a two-way algorithm for the broadcast-problem on } G \text{ and } v\}$

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

- $a_2(v, G) = \min\{\text{comm}(A) \mid A \text{ is a two-way algorithm for the accumulations-problem on } G \text{ and } v\}$
Definition

- By $\text{comm}(A)$ we denote the complexity (number of rounds) of a communication-algorithm.
- $r(G) = \min\{\text{comm}(A) \mid A \text{ is a one-way algorithm for the gossip-problem on } G\}$
- $r_2(G) = \min\{\text{comm}(A) \mid A \text{ is a two-way algorithm for the gossip-problem on } G\}$
- $b(v, G) = \min\{\text{comm}(A) \mid A \text{ is a one-way algorithm for the broadcast-problem on } G \text{ and } v\}$
- $b_2(v, G) = \min\{\text{comm}(A) \mid A \text{ is a two-way algorithm for the broadcast-problem on } G \text{ and } v\}$
- $a(v, G) = \min\{\text{comm}(A) \mid A \text{ is a one-way algorithm for the accumulations-problem on } G \text{ and } v\}$
- $a_2(v, G) = \min\{\text{comm}(A) \mid A \text{ is a two-way algorithm for the accumulations-problem on } G \text{ and } v\}$
**Definition**

- \( b(G) = \max \{ b(v, G) \mid v \in V \} \)
- \( b_2(G) = \max \{ b_2(v, G) \mid v \in V \} \)
- \( a(G) = \max \{ a(v, G) \mid v \in V \} \)
- \( a_2(G) = \max \{ a_2(v, G) \mid v \in V \} \)
- \( \min b(G) = \min \{ b(v, G) \mid v \in V \} \)
- \( \min a(G) = \min \{ a(v, G) \mid v \in V \} \)
Definition

- $b(G) = \max \{ b(v, G) \mid v \in V \}$
- $b_2(G) = \max \{ b_2(v, G) \mid v \in V \}$
- $a(G) = \max \{ a(v, G) \mid v \in V \}$
- $a_2(G) = \max \{ a_2(v, G) \mid v \in V \}$
- $\min b(G) = \min \{ b(v, G) \mid v \in V \}$
- $\min a(G) = \min \{ a(v, G) \mid v \in V \}$
Definition

- \( b(G) = \max \{ b(v, G) \mid v \in V \} \)
- \( b_2(G) = \max \{ b_2(v, G) \mid v \in V \} \)
- \( a(G) = \max \{ a(v, G) \mid v \in V \} \)
- \( a_2(G) = \max \{ a_2(v, G) \mid v \in V \} \)
- \( \min b(G) = \min \{ b(v, G) \mid v \in V \} \)
- \( \min a(G) = \min \{ a(v, G) \mid v \in V \} \)
Definition

- \( b(G) = \max\{b(v, G) \mid v \in V\} \)
- \( b_2(G) = \max\{b_2(v, G) \mid v \in V\} \)
- \( a(G) = \max\{a(v, G) \mid v \in V\} \)
- \( a_2(G) = \max\{a_2(v, G) \mid v \in V\} \)
- \( \min b(G) = \min\{b(v, G) \mid v \in V\} \)
- \( \min a(G) = \min\{a(v, G) \mid v \in V\} \)
Definition

- \( b(G) = \max\{b(v, G) \mid v \in V\} \)
- \( b_2(G) = \max\{b_2(v, G) \mid v \in V\} \)
- \( a(G) = \max\{a(v, G) \mid v \in V\} \)
- \( a_2(G) = \max\{a_2(v, G) \mid v \in V\} \)
- \( \min b(G) = \min\{b(v, G) \mid v \in V\} \)
- \( \min a(G) = \min\{a(v, G) \mid v \in V\} \)
**Definition**

- \( b(G) = \max \{ b(v, G) \mid v \in V \} \)
- \( b_2(G) = \max \{ b_2(v, G) \mid v \in V \} \)
- \( a(G) = \max \{ a(v, G) \mid v \in V \} \)
- \( a_2(G) = \max \{ a_2(v, G) \mid v \in V \} \)
- \( \min b(G) = \min \{ b(v, G) \mid v \in V \} \)
- \( \min a(G) = \min \{ a(v, G) \mid v \in V \} \)
**Definition**

- \( b(G) = \max \{ b(v, G) \mid v \in V \} \)
- \( b_2(G) = \max \{ b_2(v, G) \mid v \in V \} \)
- \( a(G) = \max \{ a(v, G) \mid v \in V \} \)
- \( a_2(G) = \max \{ a_2(v, G) \mid v \in V \} \)
- \( \min b(G) = \min \{ b(v, G) \mid v \in V \} \)
- \( \min a(G) = \min \{ a(v, G) \mid v \in V \} \)
First Results

- For each graph $G$ and $v \in V$ we have:
  - $a_2(v, G) = b_2(v, G)$
  - $a(v, G) = b(v, G)$
  - $a(G) = b(G)$
  - $\text{mina}(G) = \text{minb}(G)$
  - $b(v, G) = b_2(v, G)$
  - $b(G) = b_2(G)$

- 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:
- $a_2(v, G) = b_2(v, G)$
- $a(v, G) = b(v, G)$
- $a(G) = b(G)$
- $\text{mina}(G) = \text{minb}(G)$
- $b(v, G) = b_2(v, G)$
- $b(G) = b_2(G)$

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:
  - $a_2(v, G) = b_2(v, G)$
  - $a(v, G) = b(v, G)$
  - $a(G) = b(G)$
  - $\text{mina}(G) = \text{minb}(G)$
  - $b(v, G) = b_2(v, G)$
  - $b(G) = b_2(G)$

- 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:
  - $a_2(v, G) = b_2(v, G)$
  - $a(v, G) = b(v, G)$
  - $a(G) = b(G)$
  - $\text{mina}(G) = \text{minb}(G)$
  - $b(v, G) = b_2(v, G)$
  - $b(G) = b_2(G)$

- 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:
  - $a_2(v, G) = b_2(v, G)$
  - $a(v, G) = b(v, G)$
  - $a(G) = b(G)$
  - $\text{mina}(G) = \text{minb}(G)$
  - $b(v, G) = b_2(v, G)$
  - $b(G) = b_2(G)$

- 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:
  - $a_2(v, G) = b_2(v, G)$
  - $a(v, G) = b(v, G)$
  - $a(G) = b(G)$
  - $\text{mina}(G) = \text{minb}(G)$
  - $b(v, G) = b_2(v, G)$
  - $b(G) = b_2(G)$

- 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:
  - $a_2(v, G) = b_2(v, G)$
  - $a(v, G) = b(v, G)$
  - $a(G) = b(G)$
  - $\text{mina}(G) = \text{minb}(G)$
  - $b(v, G) = b_2(v, G)$
  - $b(G) = b_2(G)$

- 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:
  - $a_2(v, G) = b_2(v, G)$
  - $a(v, G) = b(v, G)$
  - $a(G) = b(G)$
  - $\text{mina}(G) = \text{minb}(G)$
  - $b(v, G) = b_2(v, G)$
  - $b(G) = b_2(G)$

- **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:
  - $a_2(v, G) = b_2(v, G)$
  - $a(v, G) = b(v, G)$
  - $a(G) = b(G)$
  - $\text{mina}(G) = \text{minb}(G)$
  - $b(v, G) = b_2(v, G)$
  - $b(G) = b_2(G)$

- **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:
  - $a_2(v, G) = b_2(v, G)$
  - $a(v, G) = b(v, G)$
  - $a(G) = b(G)$
  - $\text{min}(G) = \text{minb}(G)$
  - $b(v, G) = b_2(v, G)$
  - $b(G) = b_2(G)$

- 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:

- $a_2(v, G) = b_2(v, G)$
- $a(v, G) = b(v, G)$
- $a(G) = b(G)$
- $\text{mina}(G) = \text{minb}(G)$
- $b(v, G) = b_2(v, G)$
- $b(G) = b_2(G)$

**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:
  - \( a_2(v, G) = b_2(v, G) \)
  - \( a(v, G) = b(v, G) \)
  - \( a(G) = b(G) \)
  - \( \text{min}_a(G) = \text{min}_b(G) \)
  - \( b(v, G) = b_2(v, G) \)
  - \( b(G) = b_2(G) \)

  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: \( \text{min}_b(G) \leq b(G) \leq r_2(G) \leq r(G) \leq 2 \cdot r_2(G) \).
  The inequalities result from the definitions.

- \( \text{min}_b(L(n)) = \lceil n/2 \rceil \)

- 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:
  - $a_2(v, G) = b_2(v, G)$
  - $a(v, G) = b(v, G)$
  - $a(G) = b(G)$
  - $\text{mina}(G) = \text{minb}(G)$
  - $b(v, G) = b_2(v, G)$
  - $b(G) = b_2(G)$

- 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:
  - $a_2(v, G) = b_2(v, G)$
  - $a(v, G) = b(v, G)$
  - $a(G) = b(G)$
  - $\text{mina}(G) = \text{minb}(G)$
  - $b(v, G) = b_2(v, G)$
  - $b(G) = b_2(G)$

- 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:
  - $a_2(v, G) = b_2(v, G)$
  - $a(v, G) = b(v, G)$
  - $a(G) = b(G)$
  - $\text{mina}(G) = \text{minb}(G)$
  - $b(v, G) = b_2(v, G)$
  - $b(G) = b_2(G)$

- 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:
  - $a_2(v, G) = b_2(v, G)$
  - $a(v, G) = b(v, G)$
  - $a(G) = b(G)$
  - $\text{mina}(G) = \text{minb}(G)$
  - $b(v, G) = b_2(v, G)$
  - $b(G) = b_2(G)$

- 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:
  - $a_2(v, G) = b_2(v, G)$
  - $a(v, G) = b(v, G)$
  - $a(G) = b(G)$
  - $\text{mina}(G) = \text{minb}(G)$
  - $b(v, G) = b_2(v, G)$
  - $b(G) = b_2(G)$

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

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

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.
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 $b(v, G) = \min b(G) = \text{mina}(G) = z$.
- 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:

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

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 $A = E_1, E_2, \ldots E_z$ be the corresponding one-way broadcast-algorithm.
- Let $B = F_1, F_2, \ldots F_z$ be the corresponding one-way accumulation-algorithm.
- Then is $F_1, F_2, \ldots F_z, E_1, E_2, \ldots E_z$ 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 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 $n = 8$):

![Graph Diagram]

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 $n = 8$):
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 $n = 8$):

![Graph Diagram]

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

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

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

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 $b(G) = r(G)$

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
\[
\dot b(G) = r(G)
\]
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 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:
  - $b(G) \leq b(H)$.
  - $\text{minb}(G) \leq \text{minb}(H)$.
  - $\text{rad}(G) \leq \text{rad}(H)$.
  - $r(G) \leq r(H)$.
  - $r_2(G) \leq r_2(H)$.
- $\text{minb}(G) \leq (\text{deg}(G) - 1) \cdot \text{rad}(G) + 1$.
- $b(G) \leq (\text{deg}(G) - 1) \cdot \text{diam}(G) + 1$.
- $b(G) \leq \text{deg}(G) \cdot \text{rad}(G)$.
- $\text{rad}(G) \leq 2(\text{deg}(G) - 1) \cdot \text{rad}(G) + 2$.
- $r(G) \leq 2(\text{deg}(G) - 1) \cdot \text{rad}(G) + 1$.

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

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

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

\[ \text{diam}(G) = \max\{\text{dist}(u, v) | u, v \in V\} \]
\[ \text{rad}(v, G) = \max\{\text{dist}(v, x) | x \in V\} \]
\[ \text{rad}(G) = \min\{\text{rad}(v, G) | v \in V\} \]
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:
  - \( b(G) \leq b(H) \).
  - \( \text{minb}(G) \leq \text{minb}(H) \).
  - \( r(G) \leq r(H) \).
  - \( r_2(G) \leq r_2(H) \).

\[ \begin{align*}
\text{diam}(G) & = \max \{ \text{dist}(u, v) \mid u, v \in V \} \\
\text{rad}(v, G) & = \max \{ \text{dist}(v, x) \mid x \in V \} \\
\text{rad}(G) & = \min \{ \text{rad}(v, G) \mid v \in V \}
\end{align*} \]
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 \( F \subset E \). 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

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

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

- \( b(G) \leq b(H) \).
- \( \min b(G) \leq \min b(H) \).
- \( r(G) \leq r(H) \).
- \( r_2(G) \leq r_2(H) \).

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

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

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

\[
\begin{align*}
\text{diam}(G) &= \max \{ \text{dist} (u, v) \mid u, v \in V \} \\
\text{rad}(v, G) &= \max \{ \text{dist} (v, x) \mid x \in V \} \\
\text{rad}(G) &= \min \{ \text{rad}(v, G) \mid v \in V \}
\end{align*}
\]
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 \( F \subseteq E \). 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 \( F \subset E \). Then we have:
  - \( b(G) \leq b(H) \).
  - \( \text{minb}(G) \leq \text{minb}(H) \).
  - \( r(G) \leq r(H) \).
  - \( r_2(G) \leq r_2(H) \).
- \( \text{minb}(G) \leq (\text{deg}(G) - 1) \cdot \text{rad}(G) + 1 \).
- \( b(G) \leq (\text{deg}(G) - 1) \cdot \text{diam}(G) + 1 \).
- \( b(G) \leq \text{deg}(G) \cdot \text{rad}(G) \).
- \( r(G) \leq 2(\text{deg}(G) - 1) \cdot \text{rad}(G) + 2 \).
- \( r_2(G) \leq 2(\text{deg}(G) - 1) \cdot \text{rad}(G) + 1 \).
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.
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.
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.
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.
**Optimal Broadcast-Tree**

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

![Diagram of a broadcast-tree]

- 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 $T_j$. 
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 $v_0, v_1, \ldots, v_{i-1}$ 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 $v_0, v_1, \cdots, v_{i-1}$ 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 $v_0, v_1, \ldots, v_{i-1}$ 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 $T_j$. 
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 $v_0, v_1, \cdots, v_{i-1}$ 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](image)
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 \((K(n))\):
\[
\text{for } t = 1 \text{ to } \lceil \log n \rceil \text{ do } \\
\quad \text{for all } i \in \{0, 1, \ldots, 2^{t-1} - 1\} \text{ do in parallel } \\
\quad \quad \text{if } i + 2^{t-1} \leq n \text{ then } \\
\quad \quad \quad i \text{ sends to } i + 2^{t-1}
\]

Proof \((HQ(m))\):
\[
\text{for } i = 1 \text{ to } m \text{ do } \\
\quad \text{for all } a_1, a_2, \ldots, a_{i-1} \in \{0, 1\} \text{ do in parallel } \\
\quad \quad a_1 a_2 \cdots a_{i-1} 00 \cdots 0 \text{ sends to } a_1 a_2 \cdots a_{i-1} 10 \cdots 0
\]
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 ($K(n)$):

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

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

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

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

Proof ($HQ(m)$):

$\textbf{for } i = 1 \textbf{ to } m \textbf{ do}$

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

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

Lemma

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

Idea of proof:

- $b(\varepsilon, T_k(m)) = k \cdot m$.
- $b(\varepsilon, T_k(m)) \leq b(v, T_k(m))$.
- Note that $v$ 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.
Lemma

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

Idea of proof:

- $b(\varepsilon, T_k(m)) = k \cdot m$.
- $b(\varepsilon, T_k(m)) \leq b(v, T_k(m))$.
- Note that $v$ 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 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.
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.
Comlexity

**Definition:**

The special Broadcast-Problem is:

- Given: $G = (V, E)$, $v \in V$ and $k \in \mathbb{N}$.
- Question: Does $b(v, G) \leq k$ hold?

**Definition:**

The Broadcast-Problem is:

- Given: $G = (V, E)$ and $k \in \mathbb{N}$.
- Question: Does $b(G) \leq k$ hold?
Complexity

Definition:
The special Broadcast-Problem is:
- Given: $G = (V, E)$, $v \in V$ and $k \in \mathbb{N}$.
- Question: Does $b(v, G) \leq k$ hold?

Definition:
The Broadcast-Problem is:
- Given: $G = (V, E)$ and $k \in \mathbb{N}$.
- Question: Does $b(G) \leq k$ hold?
Theorem:
The special Broadcast-Problem on trees is in $\mathcal{P}$.

- The algorithm computes recursively the broadcast-time from a node (which we consider as root) in its subtree.
- For the leaves this time is 0.
- When all broadcast-times are computed for all successors of the root, we sort these times.
- After this we may compute the order of subtrees of the root in which we forward the information from the root.
- Example: 5 subtrees have broadcast-times 10, 10, 9, 9, 7. Then we inform these subtrees in the same order. The total broadcast-time from the root is $\max(10+1, 10+2, 9+3, 9+4, 7+5) = 13$.

Theorem:
The Broadcast-Problem on trees is in $\mathcal{P}$.
The special Broadcast-Problem on trees is in $\mathcal{P}$.

- The algorithm computes recursively the broadcast-time from a node (which we consider as root) in its subtree.
- For the leaves, this time is 0.
- When all broadcast-times are computed for all successors of the root, we sort these times.
- After this, we may compute the order of subtrees of the root in which we forward the information from the root.
- Example: 5 subtrees have broadcast-times 10, 10, 9, 9, 7. Then we inform these subtrees in the same order. The total broadcast-time from the root is $\max(10 + 1, 10 + 2, 9 + 3, 9 + 4, 7 + 5) = 13$.

The Broadcast-Problem on trees is in $\mathcal{P}$.
The special Broadcast-Problem on trees is in $\mathcal{P}$.

- The algorithm computes recursively the broadcast-time from a node (which we consider as root) in its subtree.
- For the leafs is this time 0.
- When all broadcast-times are computed for all successors of the root, we sort these times.
- After this we may compute the order of subtrees of the root in which we forward the information from the root.
- Example: 5 subtrees have broadcast-times 10, 10, 9, 9, 7. Then we inform these subtrees in the same order. The total broadcast-time from the root is $\max(10 + 1, 10 + 2, 9 + 3, 9 + 4, 7 + 5) = 13$. 

The Broadcast-Problem on trees is in $\mathcal{P}$. 
Theorem:
The special Broadcast-Problem on trees is in $\mathcal{P}$.

- The algorithm computes recursively the broadcast-time from a node (which we consider as root) in its subtree.
- For the leafs is this time 0.
- When all broadcast-times are computed for all successors of the root, we sort these times.
- After this we may compute the order of subtrees of the root in which we forward the information from the root.
- Example: 5 subtrees have broadcast-times 10, 10, 9, 9, 7. Then we inform these subtrees in the same order. The total broadcast-time from the root is $\max(10 + 1, 10 + 2, 9 + 3, 9 + 4, 7 + 5) = 13$.

Theorem:
The Broadcast-Problem on trees is in $\mathcal{P}$.
The special Broadcast-Problem on trees is in $\mathcal{P}$.

- The algorithm computes recursively the broadcast-time from a node (which we consider as root) in its subtree.
- For the leaves, this time is 0.
- When all broadcast-times are computed for all successors of the root, we sort these times.
- After this, we may compute the order of subtrees of the root in which we forward the information from the root.

Example: 5 subtrees have broadcast-times $10, 10, 9, 9, 7$. Then we inform these subtrees in the same order. The total broadcast-time from the root is $\max(10 + 1, 10 + 2, 9 + 3, 9 + 4, 7 + 5) = 13$.

The Broadcast-Problem on trees is in $\mathcal{P}$.
Complexity

Theorem:
The special Broadcast-Problem on trees is in $\mathcal{P}$.

- The algorithm computes recursively the broadcast-time from a node (which we consider as root) in its subtree.
- For the leafs is this time 0.
- When all broadcast-times are computed for all successors of the root, we sort these times.
- After this we may compute the order of subtrees of the root in which we forward the information from the root.

Example: 5 subtrees have broadcast-times 10, 10, 9, 9, 7. Then we inform these subtrees in the same order. The total broadcast-time from the root is $\max(10 + 1, 10 + 2, 9 + 3, 9 + 4, 7 + 5) = 13$.

Theorem:
The Broadcast-Problem on trees is in $\mathcal{P}$.
The special Broadcast-Problem on trees is in $\mathcal{P}$.

- The algorithm computes recursively the broadcast-time from a node (which we consider as root) in its subtree.
- For the leafs is this time 0.
- When all broadcast-times are computed for all successors of the root, we sort these times.
- After this we may compute the order of subtrees of the root in which we forward the information from the root.
- Example: 5 subtrees have broadcast-times 10, 10, 9, 9, 7. Then we inform these subtrees in the same order. The total broadcast-time from the root is $\max(10 + 1, 10 + 2, 9 + 3, 9 + 4, 7 + 5) = 13$. 

Theorem:
The Broadcast-Problem on trees is in $\mathcal{P}$.
Complexity

Theorem:
The special Broadcast-Problem on trees is in $\mathcal{P}$.

- The algorithm computes recursively the broadcast-time from a node (which we consider as root) in its subtree.
- For the leaves is this time 0.
- When all broadcast-times are computed for all successors of the root, we sort these times.
- After this we may compute the order of subtrees of the root in which we forward the information from the root.
- Example: 5 subtrees have broadcast-times 10, 10, 9, 9, 7. Then we inform these subtrees in the same order. The total broadcast-time from the root is $\max(10 + 1, 10 + 2, 9 + 3, 9 + 4, 7 + 5) = 13$.

Theorem:
The Broadcast-Problem on trees is in $\mathcal{P}$.
Theorem:
The special Broadcast-Problem is in $\mathcal{NP}$.

Proof: simple exercise.

- IF a message from node $v$ has to be send to node $w$ and the remaining
time is the same as the distance between $v$ and $w$, then we call this
message critical.

- I.e. the messages has to be forwarded towards $w$ without any delay.

- Is the shortest path between $v$ and $w$ unique, then we know precisely the
way (times and places) the messages has to traverse towards $w$.

- If there exists an other node $w'$ with: $\text{dist}(v, w) = \text{dist}(v, w') + 1$ and the
shortest path towards $w'$ splits from the path from $v$ to $w$, then is the
message also critical on this path.
Theorem:

The special Broadcast-Problem is in \( \mathcal{NPC} \).

Proof: simple exercise.

- **IF** a message from node \( v \) has to be send to node \( w \) and the remaining time is the same as the distance between \( v \) and \( w \), then we call this message critical.

- I.e. the messages has to be forwarded towards \( w \) without any delay.

- Is the shortest path between \( v \) and \( w \) unique, then we know precisely the way (times and places) the messages has to traverse towards \( w \).

- If there exists an other node \( w' \) with: \( \text{dist}(v, w) = \text{dist}(v, w') + 1 \) and the shortest path towards \( w' \) splits from the path from \( v \) to \( w \), then is the message also critical on this path.
Theorem:

The special Broadcast-Problem is in $\mathcal{NP}$.

Proof: simple exercise (if we have the idea).

- IF a message from node $v$ has to be send to node $w$ and the remaining time is the same as the distance between $v$ and $w$, then we call this message critical.

- I.e. the messages has to be forwarded towards $w$ without any delay.

- Is the shortest path between $v$ and $w$ unique, then we know precisely the way (times and places) the messages has to traverse towards $w$.

- If there exists an other node $w'$ with: $\text{dist}(v, w) = \text{dist}(v, w') + 1$ and the shortest path towards $w'$ splits from the path from $v$ to $w$, then is the message also critical on this path.
Complexity

Theorem:
The special Broadcast-Problem is in \( \mathcal{NP} \).

Proof: simple exercise.

- IF a message from node \( v \) has to be send to node \( w \) and the remaining time is the same as the distance between \( v \) and \( w \), then we call this message critical.

- I.e. the messages has to be forwarded towards \( w \) without any delay.

- Is the shortest path between \( v \) and \( w \) unique, then we know precisely the way (times and places) the messages has to traverse towards \( w \).

- If there exists an other node \( w' \) with: \( \text{dist}(v, w) = \text{dist}(v, w') + 1 \) and the shortest path towards \( w' \) splits from the path from \( v \) to \( w \), then is the message also critical on this path.
Complexity

Theorem:
The special Broadcast-Problem is in \(\mathcal{NP}C\).

Proof: simple exercise.

- IF a message from node \(v\) has to be send to node \(w\) and the remaining time is the same as the distance between \(v\) and \(w\), then we call this message critical.
- I.e. the messages has to be forwarded towards \(w\) without any delay.
- Is the shortest path between \(v\) and \(w\) unique, then we know precisely the way (times and places) the messages has to traverse towards \(w\).
- If there exists an other node \(w'\) with: \(\text{dist}(v, w) = \text{dist}(v, w') + 1\) and the shortest path towards \(w'\) splits from the path from \(v\) to \(w\), then is the message also critical on this path.
Complexity

Theorem:
The special Broadcast-Problem is in $\mathcal{NP\bar{C}}$.

Proof: simple exercise.

- IF a message from node $v$ has to be send to node $w$ and the remaining time is the same as the distance between $v$ and $w$, then we call this message critical.

- I.e. the messages has to be forwarded towards $w$ without any delay.

- Is the shortest path between $v$ and $w$ unique, then we know precisely the way (times and places) the messages has to traverse towards $w$.

- If there exists an other node $w'$ with: $\text{dist}(v, w) = \text{dist}(v, w') + 1$ and the shortest path towards $w'$ splits from the path from $v$ to $w$, then is the message also critical on this path.
Idea of the Proof

Broadcast from $a_0$ in 9 rounds:

Thus each node $a_i$, $b_i$ has to be informed in round $i$. 

![Diagram showing broadcast from $a_0$ in 9 rounds]
Idea of the Proof

Broadcast from $a_0$ in 9 rounds:

$\begin{align*}
\text{Broadcast from } a_0 \text{ in 9 rounds:} \\
\end{align*}$
Idea of the Proof

Broadcast from $a_0$ in 9 rounds:
Idea of the Proof

Broadcast from $a_0$ in 9 rounds:

Thus each node $a_i, b_i$ has to be informed in round $i$. 
Idea of the Proof

Broadcast from $a_0$ in 9 rounds:

Thus each node $a_i$, $b_i$ has to be informed in round $i$. 
Idea of the Proof

Broadcast from $a_0$ in 9 rounds:
Idea of the Proof

Broadcast from $a_0$ in 9 rounds:

Thus each node $a_i$, $b_i$ has to be informed in round $i$. 
Idea of the Proof

Broadcast from $a_0$ in 9 rounds:

Thus each node $a_i$, $b_i$ has to be informed in round $i$. 
Idea of the Proof

Broadcast from $a_0$ in 9 rounds:
Idea of the Proof

Broadcast from $a_0$ in 9 rounds:

Thus each node $a_i$, $b_i$ has to be informed in round $i$. 
Idea of the Proof

Broadcast from $a_0$ in 9 rounds:

Thus each node $a_i$, $b_i$ has to be informed in round $i$. 
Idea of the Proof

Broadcast from $a_0$ in 9 rounds:

$\begin{align*}
& a_0 \rightarrow a_1 \rightarrow a_2 \rightarrow a_3 \rightarrow a_4 \\
& \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \qua
Idea of the Proof

Broadcast from $a_0$ in 9 rounds:
Idea of the Proof

Broadcast from $a_0$ in 9 rounds:

Thus each node $a_i$, $b_i$ has to be informed in round $i$. 

\[
\begin{align*}
\text{broadcast from } & a_0 \\
\text{in 9 rounds:} & \\
\end{align*}
\]
Idea of the Proof

Broadcast from $a_0$ in 9 rounds:

Thus each node $a_i, b_i$ has to be informed in round $i$. 
Idea of the Proof

Broadcast from $a_0$ in 9 rounds:

```

Thus each node $a_i, b_i$ has to be informed in round $i$.
```
Idea of the Proof

Broadcast from $a_0$ in 9 rounds:

Thus each node $a_i$, $b_i$ has to be informed in round $i$. 
Idea of the Proof

Broadcast from $a_0$ in 9 rounds:

Thus each node $a_i, b_i$ has to be informed in round $i$. 
Idea of the Proof

Broadcast from $a_0$ in 9 rounds:

Thus each node $a_i$, $b_i$ has to be informed in round $i$. 
Idea of the Proof (Part A)

Broadcast from $a_0$ in 9 rounds:
Idea of the Proof (Part A)

Broadcast from $a_0$ in 9 rounds:
Idea of the Proof (Part A)

Broadcast from $a_0$ in 9 rounds:
Idea of the Proof (Part A)

Broadcast from \(a_0\) in 9 rounds:
Idea of the Proof (Part A)

Broadcast from $a_0$ in 9 rounds:
Idea of the Proof (Part A)

Broadcast from $a_0$ in 9 rounds:

- $a_0$ to $a_1$ to $a_2$
- $b_1$ to $b_2$
- $c_1$ to $c_2$
- $d_1$ to $d_2$
- $e_1$ to $e_2$

Can be extended to any number of "paths".
Idea of the Proof (Part A)

Broadcast from $a_0$ in 9 rounds:

- $a_0$ to $a_1$ to $a_2$ to $a_3$ to $a_4$ to $a_5$ to $a_6$ to $a_7$ to $a_8$ to $a_9$
- $b_2$ to $b_3$ to $b_4$ to $b_5$ to $b_6$ to $b_7$ to $b_8$ to $b_9$
- $c_3$ to $c_4$ to $c_5$ to $c_6$ to $c_7$ to $c_8$ to $c_9$
- $d_4$ to $d_6$ to $d_6$ to $d_7$ to $d_8$ to $d_9$
- $e_4$ to $e_5$ to $e_6$ to $e_7$ to $e_8$ to $e_9$

May be extended to any number of "paths".
Idea of the Proof (Part A)

Broadcast from $a_0$ in 9 rounds:
Idea of the Proof (Part A)

Broadcast from $a_0$ in 9 rounds:
Idea of the Proof (Part A)

Broadcast from $a_0$ in 9 rounds:

May be extended to any number of “paths”.
Idea of the Proof (Part A)

Broadcast from $a_0$ in 9 rounds:

May be extended to any number of “paths”.
Idea of the Proof (Part A)

Broadcast from $a_0$ in 9 rounds:

May be extended to any number of “paths”.
Idea of the Proof (Part A)

Broadcast from $a_0$ in 9 rounds:

May be extended to any number of “paths”.
Idea of the Proof (Part A)

Broadcast from $a_0$ in 9 rounds:

May be extended to any number of “paths”.
Idea of the Proof (Part A)

Broadcast from $a_0$ in 9 rounds:

May be extended to any number of “paths”.

Idea of the Proof (Part A)

Broadcast from $a_0$ in 9 rounds:

May be extended to any number of “paths”.
Idea of the Proof (Part A)

Broadcast from $a_0$ in 9 rounds:

May be extended to any number of “paths”.
Idea of the Proof (Part A)

Broadcast from $a_0$ in 9 rounds:

May be extended to any number of “paths”.
Idea of the Proof (Part A)

Broadcast from $a_0$ in 9 rounds:

May be extended to any number of “paths”.
Idea for the Variables

Consider the following situation:

- There are unique shortest paths from \( v \) to \( w, w', w'' \), which share the same splitting node.

- Assume that \( \text{dist}(v, w) - 2 = \text{dist}(v, w') = \text{dist}(v, w'') \) holds and that the message on the path from \( v \) towards \( w \) is critical.

- Then will be one of the other paths (i.e. from \( v \) to \( w' \)) critical.

- The other path (i.e. from \( v \) to \( w'' \)) is not critical:
  - We may delay the message on that path one time or
  - we may inform an additional node in the last step. informieren.

- We have now the idea for the “variable”: one path from \( v \) to \( w' \) is critical or the other path from \( v \) to \( w'' \) is critical.
Idea for the Variables

Consider the following situation:

- There are unique shortest paths from $v$ to $w, w', w''$, which share the same splitting node.
- Assume that $\text{dist}(v, w) - 2 = \text{dist}(v, w') = \text{dist}(v, w'')$ holds and that the message on the path from $v$ towards $w$ is critical.
- Then will be one of the other paths (i.e. from $v$ to $w'$) critical.
- The other path (i.e. from $v$ to $w''$) is not critical:
  - We may delay the message on that path one time or
  - we may inform an additional node in the last step. informieren.
- We have now the idea for the “variable”: one path from $v$ to $w'$ is critical or the other path from $v$ to $w''$ is critical.
Idea for the Variables

Consider the following situation:

- There are unique shortest paths from \( v \) to \( w, w', w'' \), which share the same splitting node.
- Assume that \( \text{dist}(v, w) - 2 = \text{dist}(v, w') = \text{dist}(v, w'') \) holds and that the message on the path from \( v \) towards \( w \) is critical.
- Then will be one of the other paths (i.e. from \( v \) to \( w' \)) critical.
- The other path (i.e. from \( v \) to \( w'' \)) is not critical:
  - We may delay the message on that path one time or
  - we may inform an additional node in the last step. informieren.
- We have now the idea for the “variable”: one path from \( v \) to \( w' \) is critical or the other path from \( v \) to \( w'' \) is critical.
Idea for the Variables

Consider the following situation:

- There are unique shortest paths from \( v \) to \( w, w', w'' \), which share the same splitting node.

- Assume that \( \text{dist}(v, w) - 2 = \text{dist}(v, w') = \text{dist}(v, w'') \) holds and that the message on the path from \( v \) towards \( w \) is critical.

- Then will be one of the other paths (i.e. from \( v \) to \( w' \)) critical.

- The other path (i.e. from \( v \) to \( w'' \)) is not critical:
  - We may delay the message on that path one time or
  - we may inform an additional node in the last step. informieren.

- We have now the idea for the “variable”: one path from \( v \) to \( w' \) is critical or the other path from \( v \) to \( w'' \) is critical.
Idea for the Variables

Consider the following situation:

- There are unique shortest paths from $v$ to $w, w', w''$, which share the same splitting node.
- Assume that $\text{dist}(v, w) - 2 = \text{dist}(v, w') = \text{dist}(v, w'')$ holds and that the message on the path from $v$ towards $w$ is critical.
- Then will be one of the other paths (i.e. from $v$ to $w'$) critical.
- The other path (i.e. from $v$ to $w''$) is not critical:
  - We may delay the message on that path one time or
  - we may inform an additional node in the last step. informieren.
- We have now the idea for the “variable”: one path from $v$ to $w'$ is critical or the other path from $v$ to $w''$ is critical.
Idea for the Variables

Consider the following situation:

- There are unique shortest paths from $v$ to $w, w', w''$, which share the same splitting node.
- Assume that $\text{dist}(v, w) - 2 = \text{dist}(v, w') = \text{dist}(v, w'')$ holds and that the message on the path from $v$ towards $w$ is critical.
- Then will be one of the other paths (i.e. from $v$ to $w'$) critical.
- The other path (i.e. from $v$ to $w''$) is not critical:
  - We may delay the message on that path one time or
  - we may inform an additional node in the last step. informieren.

- We have now the idea for the “variable”: one path from $v$ to $w'$ is critical or the other path from $v$ to $w''$ is critical.
Idea for the Variables

Consider the following situation:

- There are unique shortest paths from $v$ to $w, w', w''$, which share the same splitting node.
- Assume that $\text{dist}(v, w) - 2 = \text{dist}(v, w') = \text{dist}(v, w'')$ holds and that the message on the path from $v$ towards $w$ is critical.
- Then will be one of the other paths (i.e. from $v$ to $w'$) critical.
- The other path (i.e. from $v$ to $w''$) is not critical:
  - We may delay the message on that path one time or
  - we may inform an additional node in the last step. informieren.
- We have now the idea for the “variable”: one path from $v$ to $w'$ is critical or the other path from $v$ to $w''$ is critical.
Idea for the Variables

Consider the following situation:

- There are unique shortest paths from $v$ to $w, w', w''$, which share the same splitting node.
- Assume that $\text{dist}(v, w) - 2 = \text{dist}(v, w') = \text{dist}(v, w'')$ holds and that the message on the path from $v$ towards $w$ is critical.
- Then will be one of the other paths (i.e. from $v$ to $w'$) critical.
- The other path (i.e. from $v$ to $w''$) is not critical:
  - We may delay the message on that path one time or
  - we may inform an additional node in the last step. informieren.
- We have now the idea for the “variable”: one path from $v$ to $w'$ is critical or the other path from $v$ to $w''$ is critical.
Idea of the Proof (Part B)

Broadcast from $a_0$ in 9 rounds:
Idea of the Proof (Part B)

Broadcast from $a_0$ in 9 rounds:

Broadcast from $a_0$ in 9 rounds:

Thus we have a "Variable".
Idea of the Proof (Part B)

Broadcast from $a_0$ in 9 rounds:

Thus we have a "Variable".
Idea of the Proof (Part B)

Broadcast from $a_0$ in 9 rounds:

Thus we have a "Variable".
Idea of the Proof (Part B)

Broadcast from $a_0$ in 9 rounds:

Thus we have a "Variable".
Idea of the Proof (Part B)

Broadcast from $a_0$ in 9 rounds:
Idea of the Proof (Part B)

Broadcast from $a_0$ in 9 rounds:
Idea of the Proof (Part B)

Broadcast from $a_0$ in 9 rounds:
Idea of the Proof (Part B)

Broadcast from $a_0$ in 9 rounds:

Thus we have a “Variable.”
Idea of the Proof (Part B)

Broadcast from $a_0$ in 9 rounds:
Idea of the Proof (Part B)

Broadcast from $a_0$ in 9 rounds:

Thus we have a "Variable".
Idea of the Proof (Part B)

Broadcast from $a_0$ in 9 rounds:
Idea of the Proof (Part B)

Broadcast from $a_0$ in 9 rounds:
Idea of the Proof (Part B)

Broadcast from $a_0$ in 9 rounds:
Idea of the Proof (Part B)

Broadcast from $a_0$ in 9 rounds:
Idea of the Proof (Part B)

Broadcast from $a_0$ in 9 rounds:

Thus we have a "Variable".
Idea of the Proof (Part B)

Broadcast from $a_0$ in 9 rounds:

Thus we have a "Variable".
Idea of the Proof (Part B)

Broadcast from $a_0$ in 9 rounds:
Idea of the Proof (Part B)

Broadcast from $a_0$ in 9 rounds:
Idea of the Proof (Part B)

Broadcast from $a_0$ in 9 rounds:

Thus we have a “Variable.”
Idea of the Proof (Part B)

Broadcast from $a_0$ in 9 rounds:
Idea of the Proof (Part B)

Broadcast from $a_0$ in 9 rounds:

Thus we have a “Variable”.
3-SAT

Definition

A boolean formula $\mathcal{F}$ is in 3-CNF (EXACT-3-CNF):

$$\mathcal{F}(x_1, x_2, \ldots, x_r) = \bigwedge_{i=1}^{m} c_i$$

(clauses) $c_i = (l_1^i \lor l_2^i \lor l_3^i)$ $\forall 1 \leq i \leq m$

(literals) $l_j^i = \begin{cases} 
\neg x_k & \text{oder} \\
 x_k & \text{für ein } k : 1 \leq k \leq r 
\end{cases}$ $\forall 1 \leq i \leq m \forall 1 \leq j \leq 3$

An assignment is a function $W : \{x_1, x_2, \ldots, x_r\} \mapsto \{0, 1\}$.

It is NP-complete to test, if there is an assignment which satisfies $F$. 
A boolean formula $\mathcal{F}$ is in 3-CNF (EXACT-3-CNF):

$$\mathcal{F}(x_1, x_2, ..., x_r) = \bigwedge_{i=1}^{m} c_i$$

(clauses) $c_i = (l_i^1 \lor l_i^2 \lor l_i^3)$ \quad \forall 1 \leq i \leq m

(literals) $l_i^j = \begin{cases} \neg x_k \text{ oder } x_k & \text{für ein } k : 1 \leq k \leq r \\ \end{cases}$ \quad \forall 1 \leq i \leq m \quad \forall 1 \leq j \leq 3

An assignment is a function $W : \{x_1, x_2, ..., x_r\} \mapsto \{0, 1\}$.

It is NP-complete to test, if there is an assignment which satisfies $F$. 
A boolean formula $F$ is in 3-CNF (EXACT-3-CNF):

$$F(x_1, x_2, ..., x_r) = \bigwedge_{i=1}^{m} c_i$$

(clauses)  
$$c_i = (l_1^i \lor l_2^i \lor l_3^i) \quad \forall 1 \leq i \leq m$$

(literals)  
$$l_j^i = \begin{cases} 
\neg x_k & \text{oder} \\
 x_k & \text{für ein } k : 1 \leq k \leq r 
\end{cases} \quad \forall 1 \leq i \leq m \quad \forall 1 \leq j \leq 3$$

An assignment is a function $W : \{x_1, x_2, ..., x_r\} \mapsto \{0, 1\}$.

It is NP-complete to test, if there is an assignment which satisfies $F$. 
### Definition

A boolean formula $\mathcal{F}$ is in 3-CNF (EXACT-3-CNF):

$$
\mathcal{F}(x_1, x_2, \ldots, x_r) = \bigwedge_{i=1}^{m} c_i
$$

*clauses*

$$
c_i = (l_1^i \lor l_2^i \lor l_3^i) \quad \forall 1 \leq i \leq m
$$

*literals*

$$
l_i^j = \begin{cases} 
\neg x_k & \text{oder} \\
x_k & \text{für ein } k : 1 \leq k \leq r 
\end{cases} \quad \forall 1 \leq i \leq m, \quad \forall 1 \leq j \leq 3
$$

An assignment is a function $W : \{x_1, x_2, \ldots, x_r\} \mapsto \{0, 1\}$.

It is NP-complete to test, if there is an assignment which satisfies $F$. 

3-SAT
Idea of the Proof (Part C)

Thus we have many "variables".
Idea of the Proof (Part C)

Thus we have many "variables".
Idea of the Proof (Part C)

Thus we have many "variables".
Idea of the Proof (Part C)

Thus we have many "variables."
Idea of the Proof (Part C)

Thus we have many "variables".
Idea of the Proof (Part C)

Thus we have many "variables".
Idea of the Proof (Part C)

Thus we have many "variables".
Idea of the Proof (Part C)

Thus we have many "variables".
Idea of the Proof (Part C)
Idea of the Proof (Part C)
Idea of the Proof (Part C)

Thus we have many "variables."
Idea of the Proof (Part C)

Thus we have many "variables."
Idea of the Proof (Part C)

Thus we have many "variables".
Idea of the Proof (Part C)

Thus we have many "variables".
Idea of the Proof (Part C)

Thus we have many "variables".
Idea of the Proof (Part C)

Thus we have many "variables".
Idea of the Proof (Part C)
Idea of the Proof (Part C)

Thus we have many "variables".
Idea of the Proof (Part C)

Thus we have many "variables".
Idea of the Proof (Part C)

Thus we have many “variables”.
The last Step

- So far we are able to construct any number of variables.
- But the clauses are still missing.
- In 3-SAT a clause has to be satisfied by some variable.
- We may represent a clause by a node, which may only be informed the variables (paths), which are not critical (which represent the boolean value “true”). We have now the full idea for the reduction to 3-SAT.
The last Step

- So far we are able to construct any number of variables.
- But the clauses are still missing.
- In 3-SAT a clause has to be satisfied by some variable.
- We may represent a clause by a node, which may only be informed the variables (paths), which are not critical (which represent the boolean value “true”). We have now the full idea for the reduction to 3-SAT.
The last Step

- So far we are able to construct any number of variables.
- But the clauses are still missing.
- In 3-SAT a clause has to be satisfied by some variable.
- We may represent a clause by a node, which may only be informed the variables (paths), which are not critical (which represent the boolean value “true”). We have now the full idea for the reduction to 3-SAT.
The last Step

- So far we are able to construct any number of variables.
- But the clauses are still missing.
- In 3-SAT a clause has to be satisfied by some variable.
- We may represent a clause by a node, which may only be informed the variables (paths), which are not critical (which represent the boolean value "true"). We have now the full idea for the reduction to 3-SAT.
The last Step

- So far we are able to construct any number of variables.
- But the clauses are still missing.
- In 3-SAT a clause has to be satisfied by some variable.
- We may represent a clause by a node, which may only be informed the variables (paths), which are not critical (which represent the boolean value “true”). We have now the full idea for the reduction to 3-SAT.
Idea of the Proof (Part D)

Thus we have a "clause".
Idea of the Proof (Part D)

Thus we have a "clause".
Idea of the Proof (Part D)

Thus we have a "clause".
Idea of the Proof (Part D)

Thus we have a "clause".
Idea of the Proof (Part D)

Thus we have a “clause.”
Idea of the Proof (Part D)

Thus we have a “clause”.
Idea of the Proof (Part D)

Thus we have a “clause”.
Thus we have a "clause".
Idea of the Proof (Part D)

Thus we have a “clause”.
Thus we have a “clause”.
Thus we have a “clause”.
Idea of the Proof (Part D)

Thus we have a “clause”.
Idea of the Proof (Part D)

Thus we have a “clause”.
Idea of the Proof (Part D)

Thus we have a “clause”.
Idea of the Proof (Part D)

Thus we have a “clause”.
Idea of the Proof (Part D)

Thus we have a “clause”.
Idea of the Proof (Part D)

Thus we have a “clause”.
Idea of the Proof (Part D)

Thus we have a “clause”.
Idea of the Proof (Part D)

Thus we have a “clause”.

Thus we have a “clause”.
Idea of the Proof

- Consider a boolean formula $\mathcal{F}$ from 3-SAT:
  - Generate for each of the $n$ variables from $\mathcal{F}$ a critical path (Part A).
  - Generate for each of the above critical paths an alternative (Part B).
  - Thus we have now all literals.
  - Generate for each literal $x$ paths, if the literal occurs in $\mathcal{F}$ $x$ times (Part C).
  - Generate for each clause a construction given by Part D.
Idea of the Proof

- Consider a boolean formula $\mathcal{F}$ from 3-SAT:
  - Generate for each of the $n$ variables from $\mathcal{F}$ a critical path (Part A).
  - Generate for each of the above critical paths an alternative (Part B).
  - Thus we have now all literals.
  - Generate for each literal $x$ paths, if the literal occurs in $\mathcal{F}$ $x$ times (Part C).
  - Generate for each clause a construction given by Part D.
Idea of the Proof

- Consider a boolean formula $\mathcal{F}$ from 3-$SAT$:
- Generate for each of the $n$ variables from $\mathcal{F}$ a critical path (Part A).
- Generate for each of the above critical paths an alternative (Part B).
- Thus we have now all literals.
- Generate for each literal $x$ paths, if the literal occurs in $\mathcal{F}$ $x$ times (Part C).
- Generate for each clause a construction given by Part D.
Idea of the Proof

- Consider a boolean formula $\mathcal{F}$ from 3-SAT:
- Generate for each of the $n$ variables from $\mathcal{F}$ a critical path (Part A).
- Generate for each of the above critical paths an alternative (Part B).
- Thus we have now all literals.
- Generate for each literal $x$ paths, if the literal occurs in $\mathcal{F}$ $x$ times (Part C).
- Generate for each clause a construction given by Part D.
Idea of the Proof

- Consider a boolean formula $\mathcal{F}$ from 3-$SAT$:
  
- Generate for each of the $n$ variables from $\mathcal{F}$ a critical path (Part A).
  
- Generate for each of the above critical paths an alternative (Part B).
  
- Thus we have now all literals.
  
- Generate for each literal $x$ paths, if the literal occurs in $\mathcal{F}$ $x$ times (Part C).
  
- Generate for each clause a construction given by Part D.
Idea of the Proof

- Consider a boolean formula \( \mathcal{F} \) from \( 3-SAT \):
- Generate for each of the \( n \) variables from \( \mathcal{F} \) a critical path (Part A).
- Generate for each of the above critical paths an alternative (Part B).
- Thus we have now all literals.
- Generate for each literal \( x \) paths, if the literal occurs in \( \mathcal{F} \times \) times (Part C).
- Generate for each clause a construction given by Part D.
Idea of the Proof

- Consider a boolean formula $\mathcal{F}$ from 3-SAT:
- Generate for each of the $n$ variables from $\mathcal{F}$ a critical path (Part A).
- Generate for each of the above critical paths an alternative (Part B).
- Thus we have now all literals.
- Generate for each literal $x$ paths, if the literal occurs in $\mathcal{F}$ $x$ times (Part C).
- Generate for each clause a construction given by Part D.
Complexity

**Theorem:**

The special broadcast-problem on graphs of degree 3 is in $\mathcal{NP}_C$.

Proof: it is easy to build the above construction with nodes of degree $\leq 3$.

**Theorem:**

The special broadcast-problem on planar graphs of degree 3 is in $\mathcal{NP}_C$.

Idea of proof: The planar 3-SAT is in $\mathcal{NP}_C$. That is the dependency graph between clauses and variables is planar.

**Definition:**

Let $\mathcal{F}$ be a boolean formula in KNF. Let $V$ be the variables and $C$ be the clauses. The dependency graph is:

$$G_{\mathcal{F}} = (V, C, \{\{v, c\} \mid v \text{ is in } c\})$$
Theorem:
The special broadcast-problem on graphs of degree 3 is in \( \mathcal{NPC} \).

Proof: it is easy to build the above construction with nodes of degree \( \leq 3 \).

Theorem:
The special broadcast-problem on planar graphs of degree 3 is in \( \mathcal{NPC} \).

Idea of proof: The planar 3-SAT is in \( \mathcal{NPC} \). That is the dependency graph between clauses and variables is planar.

Definition:
Let \( \mathcal{F} \) be a boolean formula in \( \text{KNF} \). Let \( V \) be the variables and \( C \) be the clauses. The dependency graph is:

\[
G_{\mathcal{F}} = (V, C, \{\{v, c\} \mid v \text{ is in } c\})
\]
Theorem:
The special broadcast-problem on graphs of degree 3 is in $\mathcal{NP}$.

Proof: it is easy to build the above construction with nodes of degree $\leq 3$.

Theorem:
The special broadcast-problem on planar graphs of degree 3 is in $\mathcal{NP}$.

Idea of proof: The planar 3-SAT is in $\mathcal{NP}$. That is the dependency graph between clauses and variables is planar.

Definition:
Let $\mathcal{F}$ be a boolean formula in KNF. Let $V$ be the variables and $C$ be the clauses. The dependency graph is:

$$G_{\mathcal{F}} = (V, C, \{\{v, c\} \mid v \text{ is in } c\})$$
Complexity

Theorem:
The special broadcast-problem on graphs of degree 3 is in $\mathcal{NP}$.  
Proof: it is easy to build the above construction with nodes of degree $\leq 3$.

Theorem:
The special broadcast-problem on planar graphs of degree 3 is in $\mathcal{NP}$.  
Idea of proof: The planar 3-SAT is in $\mathcal{NP}$. That is the dependency graph between clauses and variables is planar.

Definition:
Let $F$ be a boolean formula in $KNF$. Let $V$ be the variables and $C$ be the clauses. The dependency graph is:

$$G_F = (V, C, \{\{v, c\} \mid v \text{ is in } c\})$$
**Theorem:**

The broadcast-problem on planar graphs of degree 3 is in $\mathcal{NP}$.  

**Proof:**

- Extend the above construction, such that there is a unique “hardest” node.
- Add to the above construction a very long path.
- Thus the broadcast from the start node of the long path is the hardest.
Complexity

Theorem:
The broadcast-problem on planar graphs of degree 3 is in $\mathcal{NP}$.

Proof:

- Extend the above construction, such that there is a unique "hardest" node.
- Add to the above construction a very long path.
- Thus the broadcast from the start node of the long path is the hardest.
Complexity

Theorem:
The broadcast-problem on planar graphs of degree 3 is in $\mathcal{NP}$.

Proof:

- Extend the above construction, such that there is a unique “hardest” node.
- **Add to the above construction a very long path.**
- Thus the broadcast from the start node of the long path is the hardest.
Theorem:
The broadcast-problem on planar graphs of degree 3 is in \( \mathcal{NP} \).

Proof:

- Extend the above construction, such that there is a unique “hardest” node.
- Add to the above construction a very long path.
- Thus the broadcast from the start node of the long path is the hardest.
Complexity

Theorem:
The broadcast-problem on planar graphs of degree 3 is in \( \mathcal{NP} \).

Proof:

- Extend the above construction, such that there is a unique “hardest” node.
- Add to the above construction a very long path.
- Thus the broadcast from the start node of the long path is the hardest.
**Definition:**

The gossip-problem is:

- **Given:** \( G = (V, E) \) and \( k \in \mathbb{N} \).
- **Question:** Does \( r_2(G) \leq k \) hold?

**Theorem:**

The gossip-problem is in \( \mathcal{NP} \).

Proof: Extend the above construction, such that there is a unique “hardest” node.
Complexity

**Definition:**

The gossip-problem is:

- Given: \( G = (V, E) \) and \( k \in \mathbb{N} \).
- Question: Does \( r_2(G) \leq k \) hold?

**Theorem:**

The gossip-problem is in \( \mathcal{NPC} \).

**Proof:** Extend the above construction, such that there is a unique “hardest” node.
Definition:
The gossip-problem is:
- Given: \( G = (V, E) \) and \( k \in \mathbb{N} \).
- Question: Does \( r_2(G) \leq k \) hold?

Theorem:
The gossip-problem is in \( \mathcal{NP} \).

Proof: Extend the above construction, such that there is a unique “hardest” node.
The one-way gossip-problem is:
- Given: $G = (V, E)$ and $k \in \mathbb{N}$.
- Question: Does $r(G) \leq k$ hold?

The one-way gossip-problem is in $\mathcal{NPC}$.

Proof: Extend the above construction, such that there is a unique “hardest” node.
And prevent the blocking of critical messages.
The one-way gossip-problem is:

- Given: $G = (V, E)$ and $k \in \mathbb{N}$.
- Question: Does $r(G) \leq k$ hold?

The one-way gossip-problem is in $\mathcal{NPC}$.

Proof: Extend the above construction, such that there is a unique “hardest” node.
And prevent the blocking of critical messages.
Definition:
The one-way gossip-problem is:
- Given: \( G = (V, E) \) and \( k \in \mathbb{N} \).
- Question: Does \( r(G) \leq k \) hold?

Theorem:
The one-way gossip-problem is in \( \mathcal{NP} \).

Proof: Extend the above construction, such that there is a unique “hardest” node.
And prevent the blocking of critical messages.
Definition:
The one-way gossip-problem is:
- Given: $G = (V, E)$ and $k \in \mathbb{N}$.
- Question: Does $r(G) \leq k$ hold?

Theorem:
The one-way gossip-problem is in $\mathcal{NPC}$.

Proof: Extend the above construction, such that there is a unique “hardest” node.
And prevent the blocking of critical messages.
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:

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

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

- \( b(G) \leq (\text{deg}(G) - 1) \cdot \text{diam}(G) + 1 \)
- \( b(G) \leq \text{deg}(G) \cdot \text{rad}(G) \).
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: $\lceil \frac{5k}{2} \rceil - 2 \leq \min b(\text{CCC}(k)) = b(\text{CCC}(k)) \leq \lceil \frac{5k}{2} \rceil - 1$.

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

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

- 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.
CCC, Proof $\minb(CCC(k)) \geq \lceil 5 \cdot k/2 \rceil - 2$

- $\diam(CCC(k)) = \lfloor 5/2 \cdot k \rfloor - 2$
- 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 holds, because the CCC is node-symmetric.
**CCC, Proof** \( \text{minb}(\text{CCC}(k)) \geq \left\lceil 5 \cdot k/2 \right\rceil - 2 \)

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

- \( \text{diam}(CCC(k)) = \lceil 5/2 \cdot k \rceil - 2 \)
- 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 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 \((0, 00 \cdots 0)\) 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 \( \minb(\text{CCC}(k)) \geq \lceil 5 \cdot k/2 \rceil - 2 \)

- \( \text{diam(CCC}(k) = \lceil 5/2 \cdot k \rceil - 2 \)
- The statement holds for even \( k \).
- Let \( k \) be odd.
- Let \((0,00\cdots0)\) be the origin of the message.
- The nodes \((\lceil k/2 \rceil, 11\cdots1)\) and \((\lceil k/2 \rceil + 1, 11\cdots1)\) 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.
\[ \text{CCC, Proof } \minb(\text{CCC}(k)) \geq \left\lfloor \frac{5 \cdot k}{2} \right\rfloor - 2 \]

- \( \text{diam}(\text{CCC}(k)) = \left\lfloor \frac{5}{2} \cdot k \right\rfloor - 2 \)
- 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.
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: \( \min b(CCC(k)) = b(CCC(k)) \leq \lceil 5 \cdot k/2 \rceil - 2. \)

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.
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.
Theorem:

We have: \( \min_b(\text{SE}(k)) = b(\text{SE}(k)) = 2 \cdot k - 1 \)

Proof:

- The diameter provides the lower bound.
- Note \( \text{SE}(k) \) is not node-symmetric.
- We have to provide an algorithm for any node \( v \).
- Algorithm has to be without conflicts.
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.
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.
For each $w = a_1 a_2 \ldots a_k \in \{0, 1\}^k$, let

- $w_1 = a_1$ and
- $w(t) = a_t a_{t+1} \ldots a_k$ (for $1 \leq t \leq k$)
- $w(k + 1) = \varepsilon$.

Let $\alpha = a_1 a_2 \ldots a_k$ in $SE_k$ be the origin.

$\alpha = a_1 a_2 \ldots a_{k-1}a_k$ sends to $a_1 a_2 \ldots a_{k-1} \overline{a}_k$ (exchange);

for $t = 1$ to $k - 1$ do
  for all $\beta \in \{0, 1\}^t$ do in parallel
    begin
      if $\alpha(t) \notin \{\beta_1\}^+$
        then $\alpha(t) \beta$ sends to $\alpha(t + 1) \beta a_t$ (shuffle);
        $\alpha(t + 1) \beta a_t$ sends to $\alpha(t + 1) \beta \overline{a}_t$ (exchange)
    end;
For each \( w = a_1 a_2 \ldots a_k \in \{0, 1\}^k \), let

- \( w_1 = a_1 \) and
- \( w(t) = a_t a_{t+1} \ldots a_k \) (for \( 1 \leq t \leq k \))
- \( w(k + 1) = \varepsilon \).
- Let \( \alpha = a_1 a_2 \ldots a_k \) in \( SE_k \) be the origin.

\( \alpha = a_1 a_2 \ldots a_{k-1} a_k \) sends to \( a_1 a_2 \ldots a_{k-1} \bar{a}_k \) (exchange);

for \( t = 1 \) to \( k - 1 \) do
  for all \( \beta \in \{0, 1\}^t \) do in parallel
  begin
    if \( \alpha(t) \not\in \{\beta_1\}^+ \)
      then \( \alpha(t) \beta \) sends to \( \alpha(t + 1) \beta a_t \) (shuffle);
      \( \alpha(t + 1) \beta a_t \) sends to \( \alpha(t + 1) \beta \bar{a}_t \) (exchange)
  end;
For each \( w = a_1 a_2 \ldots a_k \in \{0, 1\}^k \), let

- \( w_1 = a_1 \) and
- \( w(t) = a_t a_{t+1} \ldots a_k \) (for \( 1 \leq t \leq k \))
- \( w(k+1) = \varepsilon \).

Let \( \alpha = a_1 a_2 \ldots a_k \) in \( SE_k \) be the origin.

\( \alpha = a_1 a_2 \ldots a_{k-1} a_k \) sends to \( a_1 a_2 \ldots a_{k-1} \bar{a}_k \) (exchange);

for \( t = 1 \) to \( k - 1 \) do

for all \( \beta \in \{0, 1\}^t \) do in parallel

begin

if \( \alpha(t) \not\in \{\beta_1\}^+ \)

then \( \alpha(t) \beta \) sends to \( \alpha(t + 1) \beta a_t \) (shuffle);

\( \alpha(t + 1) \beta a_t \) sends to \( \alpha(t + 1) \beta \bar{a}_t \) (exchange)

end;
SE, Proof

For each $w = a_1a_2 \ldots a_k \in \{0,1\}^k$, let

- $w_1 = a_1$ and
- $w(t) = a_ta_{t+1} \ldots a_k$ (for $1 \leq t \leq k$)
- $w(k+1) = \varepsilon$.

Let $\alpha = a_1a_2 \ldots a_k$ in $SE_k$ be the origin. $\alpha = a_1a_2 \ldots a_k - 1 a_k$ sends to $a_1a_2 \ldots a_{k-1} \bar{a}_k$ (exchange);

for $t = 1$ to $k-1$ do

    for all $\beta \in \{0,1\}^t$ do in parallel

    begin

    if $\alpha(t) \notin \{\beta_1\}^+$

        then $\alpha(t)\beta$ sends to $\alpha(t+1)\beta a_t$ (shuffle);

        $\alpha(t+1)\beta a_t$ sends to $\alpha(t+1)\beta \bar{a}_t$ (exchange)

    end;
For each $w = a_1a_2 \ldots a_k \in \{0, 1\}^k$, let

- $w_1 = a_1$ and
- $w(t) = a_t a_{t+1} \ldots a_k$ (for $1 \leq t \leq k$)
- $w(k + 1) = \varepsilon$.

Let $\alpha = a_1a_2 \ldots a_k$ in $SE_k$ be the origin.

$\alpha = a_1a_2 \ldots a_{k-1}a_k$ sends to $a_1a_2 \ldots a_{k-1}\overline{a}_k$ (exchange);

for $t = 1$ to $k - 1$ do

for all $\beta \in \{0, 1\}^t$ do in parallel

begin

if $\alpha(t) \not\in \{\beta_1\}^+$

then $\alpha(t)\beta$ sends to $\alpha(t + 1)\beta a_t$ (shuffle);

$\alpha(t + 1)\beta a_t$ sends to $\alpha(t + 1)\beta \overline{a}_t$ (exchange)

end;
SE, Proof

For each $w = a_1a_2 \ldots a_k \in \{0, 1\}^k$, let

- $w_1 = a_1$ and
- $w(t) = a_ta_{t+1} \ldots a_k$ (for $1 \leq t \leq k$)
- $w(k+1) = \varepsilon$.

Let $\alpha = a_1a_2 \ldots a_k$ in $SE_k$ be the origin.

$\alpha = a_1a_2 \ldots a_{k-1}a_k$ sends to $a_1a_2 \ldots a_{k-1} \bar{a}_k$ (exchange);

for $t = 1$ to $k - 1$ do
  for all $\beta \in \{0, 1\}^t$ do in parallel
  begin
    if $\alpha(t) \notin \{\beta_1\}^+$
      then $\alpha(t)\beta$ sends to $\alpha(t+1)\beta a_t$ (shuffle);
      $\alpha(t+1)\beta a_t$ sends to $\alpha(t+1)\beta \bar{a}_t$ (exchange)
  end;
SE, Proof

\[ \alpha = a_1a_2 \ldots a_{k-1}a_k \text{ sends to } a_1a_2 \ldots a_{k-1}\bar{a}_k \text{ (exchange)}; \]

for \( t = 1 \) to \( k - 1 \) do

for all \( \beta \in \{0, 1\}^t \) do in parallel begin

if \( \alpha(t) \not\in \{\beta_1\}^+ \)

then \( \alpha(t)\beta \) sends to \( \alpha(t + 1)\beta a_t \) (shuffle); \n\( \alpha(t + 1)\beta a_t \) sends to \( \alpha(t + 1)\beta \bar{a}_t \) (exchange) end;

Show: There are no conflicts!

- There is no conflict for the exchange-edges, because the last bit gives a unique sender and receiver.
- Assume there is a conflict by the shuffle-edges.
- We have \( \alpha(t)\beta = \alpha(t + 1)\gamma a_t \) for some \( \beta, \gamma \in \{0, 1\}^t \).
- Then we have: \( a_t\alpha(t + 1) = \alpha(t + 1)\gamma_1 \Rightarrow a_t = a_{t+1} = \cdots = a_k = \gamma_1 \Rightarrow \alpha(t) \in \{\gamma_1\}^+ \).
- This is a contradiction: shuffle-edges for \( \alpha(t) \in \{\gamma_1\}^+ \) are not used.
SE, Proof

\[
\alpha = a_1 a_2 \ldots a_{k-1} a_k \text{ sends to } a_1 a_2 \ldots a_{k-1} \bar{a}_k \text{ (exchange)};
\]

for \( t = 1 \) to \( k - 1 \) do

for all \( \beta \in \{0, 1\}^t \) do in parallel begin

if \( \alpha(t) \not\in \{\beta_1\}^+ \)

then \( \alpha(t)\beta \) sends to \( \alpha(t + 1)\beta a_t \) (shuffle);

\( \alpha(t + 1)\beta a_t \) sends to \( \alpha(t + 1)\beta \bar{a}_t \) (exchange) end;

Show: There are no conflicts!

- There is no conflict for the exchange-edges, because the last bit give a unique sender and receiver.
- Assume there is a conflict by the shuffle-edges.
- We have \( \alpha(t)\beta = \alpha(t + 1)\gamma a_t \) for some \( \beta, \gamma \in \{0, 1\}^t \).
- Then we have:
  \[
  a_t \alpha(t + 1) = \alpha(t + 1)\gamma_1 \Rightarrow a_t = a_{t+1} = \cdots = a_k = \gamma_1 \Rightarrow \alpha(t) \in \{\gamma_1\}^+.
  \]
- This is a contradiction: shuffle-edges for \( \alpha(t) \in \{\gamma_1\}^+ \) are not used.
\[ \alpha = a_1 a_2 \ldots a_{k-1} a_k \text{ sends to } a_1 a_2 \ldots a_{k-1} \overline{a}_k \text{ (exchange)}; \]

for \( t = 1 \) to \( k - 1 \) do

\hspace{1em} for all \( \beta \in \{0, 1\}^t \) do in parallel begin

\hspace{3em} if \( \alpha(t) \notin \{\beta_1\}^+ \)

\hspace{5em} then \( \alpha(t)\beta \) sends to \( \alpha(t + 1)\beta a_t \) (shuffle);

\hspace{5em} \( \alpha(t + 1)\beta a_t \) sends to \( \alpha(t + 1)\beta \overline{a}_t \) (exchange) end;

Show: There are no conflicts!

- There is no conflict for the exchange-edges, because the last bit give a unique sender and receiver.

- **Assume there is a conflict by the shuffle-edges.**

- We have \( \alpha(t)\beta = \alpha(t + 1)\gamma a_t \) for some \( \beta, \gamma \in \{0, 1\}^t \).

- Then we have:

\[ a_t \alpha(t + 1) = \alpha(t + 1)\gamma_1 \Rightarrow a_t = a_{t+1} = \cdots = a_k = \gamma_1 \Rightarrow \alpha(t) \in \{\gamma_1\}^+ . \]

- This is a contradiction: shuffle-edges for \( \alpha(t) \in \{\gamma_1\}^+ \) are not used.
SE, Proof

\[ \alpha = a_1a_2 \ldots a_{k-1}a_k \text{ sends to } a_1a_2 \ldots a_{k-1}\overline{a}_k \text{ (exchange)}; \]

for \( t = 1 \) to \( k - 1 \) do

for all \( \beta \in \{0, 1\}^t \) do in parallel begin

if \( \alpha(t) \not\in \{\beta \}^+ \)

then \( \alpha(t)\beta \text{ sends to } \alpha(t + 1)\beta a_t \text{ (shuffle)}; \)

\( \alpha(t + 1)\beta a_t \text{ sends to } \alpha(t + 1)\beta\overline{a}_t \text{ (exchange)} \) end;

Show: There are no conflicts!

- There is no conflict for the exchange-edges, because the last bit gives a unique sender and receiver.
- Assume there is a conflict by the shuffle-edges.
- We have \( \alpha(t)\beta = \alpha(t + 1)\gamma a_t \) for some \( \beta, \gamma \in \{0, 1\}^t \).
- Then we have:
  \[ a_t\alpha(t + 1) = \alpha(t + 1)\gamma_1 \Rightarrow a_t = a_{t+1} = \cdots = a_k = \gamma_1 \Rightarrow \alpha(t) \in \{\gamma_1\}^+. \]
- This is a contradiction: shuffle-edges for \( \alpha(t) \in \{\gamma_1\}^+ \) are not used.
α = a₁a₂...aₖ⁻₁aₖ sends to a₁a₂...aₖ⁻₁āₖ (exchange);
for t = 1 to k − 1 do
    for all β ∈ {0, 1}ᵗ do in parallel begin
        if α(t) ∉ {β₁}⁺
            then α(t)β sends to α(t + 1)βaₜ (shuffle);
        α(t + 1)βaₜ sends to α(t + 1)βāₜ (exchange) end;
Show: There are no conflicts!

- There is no conflict for the exchange-edges, because the last bit give a unique sender and receiver.
- Assume there is a conflict by the shuffle-edges.
- We have α(t)β = α(t + 1)γaₜ for some β, γ ∈ {0, 1}ᵗ.
- Then we have:
  aₜα(t + 1) = α(t + 1)γ₁ ⇒ aₜ = aₜ₊₁ = ... = aₖ = γ₁ ⇒ α(t) ∈ {γ₁}⁺.
- This is a contradiction: shuffle-edges for α(t) ∈ {γ₁}⁺ are not used.
SE, Proof

\[ \alpha = a_1 a_2 \ldots a_{k-1} a_k \text{ sends to } a_1 a_2 \ldots a_{k-1} \bar{a}_k \text{ (exchange)}; \]

for \( t = 1 \) to \( k - 1 \) do

for all \( \beta \in \{0, 1\}^t \) do in parallel begin

if \( \alpha(t) \notin \{\beta_1\}^+ \)

then \( \alpha(t) \beta \) sends to \( \alpha(t + 1) \beta a_t \) (shuffle);
\( \alpha(t + 1) \beta a_t \) sends to \( \alpha(t + 1) \beta \bar{a}_t \) (exchange) end;

Show: There are no conflicts!

- There is no conflict for the exchange-edges, because the last bit give a unique sender and receiver.
- Assume there is a conflict by the shuffle-edges.
- We have \( \alpha(t) \beta = \alpha(t + 1) \gamma a_t \) for some \( \beta, \gamma \in \{0, 1\}^t \).
- Then we have:
  \[ a_t \alpha(t + 1) = \alpha(t + 1) \gamma_1 \Rightarrow a_t = a_{t+1} = \cdots = a_k = \gamma_1 \Rightarrow \alpha(t) \in \{\gamma_1\}^+. \]
- This is a contradiction: shuffle-edges for \( \alpha(t) \in \{\gamma_1\}^+ \) are not used.
**SE, Proof**

\[ \alpha = a_1 a_2 \ldots a_{k-1} a_k \text{ sends to } a_1 a_2 \ldots a_{k-1} \overline{a}_k \text{ (exchange)}; \]

for \( t = 1 \) to \( k - 1 \) do

  for all \( \beta \in \{0, 1\}^t \) do in parallel begin

    if \( \alpha(t) \notin \{\beta_1\}^+ \)

      then \( \alpha(t) \beta \) sends to \( \alpha(t + 1) \beta a_t \) (shuffle);

      \( \alpha(t + 1) \beta a_t \) sends to \( \alpha(t + 1) \beta \overline{a}_t \) (exchange) end;

Show: There are no conflicts!

- There is no conflict for the exchange-edges, because the last bit give a unique sender and receiver.
- Assume there is a conflict by the shuffle-edges.
- We have \( \alpha(t) \beta = \alpha(t + 1) \gamma a_t \) for some \( \beta, \gamma \in \{0, 1\}^t \).
- Then we have:
  \[ a_t \alpha(t + 1) = \alpha(t + 1) \gamma_1 \Rightarrow a_t = a_{t+1} = \cdots = a_k = \gamma_1 \Rightarrow \alpha(t) \in \{\gamma_1\}^+. \]
- This is a contradiction: shuffle-edges for \( \alpha(t) \in \{\gamma_1\}^+ \) are not used.
SE, Proof

\[ \alpha = a_1 a_2 \ldots a_{k-1} a_k \text{ sends to } a_1 a_2 \ldots a_{k-1} \bar{a}_k \text{ (exchange);} \]

for \( t = 1 \) to \( k - 1 \) do

for all \( \beta \in \{0, 1\}^t \) do in parallel begin

if \( \alpha(t) \not\in \{\beta_1\}^+ \)

then \( \alpha(t)\beta \) sends to \( \alpha(t + 1)\beta a_t \) (shuffle);

\( \alpha(t + 1)\beta a_t \) sends to \( \alpha(t + 1)\beta \bar{a}_t \) (exchange) end;

Show: All nodes are informed!

- Show by induction: After \( 2 \cdot r + 1 \) rounds are all nodes \( \alpha(r + 2)\beta, \beta \in \{0, 1\}^{r+1} \) informed.
- IS: \( r = 0 \) is obvious.
- All nodes \( \alpha(r + 1) \not\in \{\beta_1\}^+, \beta \in \{0, 1\}^{r+1} \) will be informed, because all nodes \( \alpha(r + 2)\beta \) have already received the information.
- If \( \alpha(r + 1) \in \{\beta_1\}^+, \beta \in \{0, 1\}^{r+1} \) holds, then we have \( \alpha(r + 2)\beta a_{r+1} = \alpha(r + 1)\beta_1 \beta a_{r+1} \).
- This node has been informed before.
\[ \alpha = a_1 a_2 \ldots a_{k-1} a_k \text{ sends to } a_1 a_2 \ldots a_{k-1} \bar{a}_k \text{ (exchange)}; \]

for \( t = 1 \) to \( k - 1 \) do

for all \( \beta \in \{0, 1\}^t \) do in parallel begin

if \( \alpha(t) \notin \{\beta_1\}^+ \)

then \( \alpha(t)\beta \) sends to \( \alpha(t + 1)\beta a_t \) (shuffle);

\( \alpha(t + 1)\beta a_t \) sends to \( \alpha(t + 1)\beta \bar{a}_t \) (exchange) end;

Show: All nodes are informed!

- Show by induction: After \( 2 \cdot r + 1 \) rounds are all nodes \( \alpha(r + 2)\beta, \beta \in \{0, 1\}^{r+1} \) informed.
- IS: \( r = 0 \) is obvious.
- All nodes \( \alpha(r + 1) \notin \{\beta_1\}^+, \beta \in \{0, 1\}^{r+1} \) will be informed, because all nodes \( \alpha(r + 2)\beta \) have already received the information.

- If \( \alpha(r + 1) \in \{\beta_1\}^+, \beta \in \{0, 1\}^{r+1} \) holds, then we have \( \alpha(r + 2)\beta a_{r+1} = \alpha(r + 1)\beta_1 \beta a_{r+1} \).

- This node has been informed before.
SE, Proof

\[ \alpha = a_1a_2 \ldots a_{k-1}a_k \text{ sends to } a_1a_2 \ldots a_{k-1}\overline{a}_k \text{ (exchange);} \]

for \( t = 1 \) to \( k - 1 \) do

\[ \text{for all } \beta \in \{0, 1\}^t \text{ do in parallel begin} \]

\[ \text{if } \alpha(t) \not\in \{\beta_1\}^+ \]

\[ \text{then } \alpha(t)\beta \text{ sends to } \alpha(t + 1)\beta a_t \text{ (shuffle);} \]

\[ \alpha(t + 1)\beta a_t \text{ sends to } \alpha(t + 1)\beta \overline{a}_t \text{ (exchange)} \text{ end;} \]

Show: All nodes are informed!

- Show by induction: After \( 2 \cdot r + 1 \) rounds are all nodes \( \alpha(r + 2)\beta, \beta \in \{0, 1\}^{r+1} \) informed.

- IS: \( r = 0 \) is obvious.

- All nodes \( \alpha(r + 1) \not\in \{\beta_1\}^+, \beta \in \{0, 1\}^{r+1} \) will be informed, because all nodes \( \alpha(r + 2)\beta \) have already received the information.

- If \( \alpha(r + 1) \in \{\beta_1\}^+, \beta \in \{0, 1\}^{r+1} \) holds, then we have \( \alpha(r + 2)\beta a_{r+1} = \alpha(r + 1)\beta_1\beta a_{r+1} \).

- This node has been informed before.
SE, Proof

\[ \alpha = a_1 a_2 \ldots a_{k-1} a_k \text{ sends to } a_1 a_2 \ldots a_{k-1} \bar{a}_k \text{ (exchange);} \]

for \( t = 1 \) to \( k - 1 \) do

for all \( \beta \in \{0, 1\}^t \) do in parallel begin

if \( \alpha(t) \not\in \{\beta_1\}^+ \)

then \( \alpha(t)\beta \) sends to \( \alpha(t + 1)\beta a_t \) (shuffle);

\( \alpha(t + 1)\beta a_t \) sends to \( \alpha(t + 1)\beta \bar{a}_t \) (exchange) end;

Show: All nodes are informed!

- Show by induction: After \( 2 \cdot r + 1 \) rounds are all nodes \( \alpha(r + 2)\beta, \beta \in \{0, 1\}^{r+1} \) informed.
- IS: \( r = 0 \) is obvious.
- All nodes \( \alpha(r + 1) \not\in \{\beta_1\}^+, \beta \in \{0, 1\}^{r+1} \) will be informed, because all nodes \( \alpha(r + 2)\beta \) have already received the information.
- If \( \alpha(r + 1) \in \{\beta_1\}^+, \beta \in \{0, 1\}^{r+1} \) holds, then we have \( \alpha(r + 2)\beta a_{r+1} = \alpha(r + 1)\beta_1\beta a_{r+1} \).
- This node has been informed before.
SE, Proof

\[ \alpha = a_1a_2 \ldots a_{k-1}a_k \text{ sends to } a_1a_2 \ldots a_{k-1}\overline{a}_k \text{ (exchange)}; \]

for \( t = 1 \) to \( k - 1 \) do

for all \( \beta \in \{0, 1\}^t \) do in parallel begin

if \( \alpha(t) \not\in \{\beta_1\}^+ \)

then \( \alpha(t)\beta \text{ sends to } \alpha(t + 1)\beta a_t \text{ (shuffle)}; \)

\( \alpha(t + 1)\beta a_t \text{ sends to } \alpha(t + 1)\beta \overline{a}_t \text{ (exchange)} \)
end;

Show: All nodes are informed!

- Show by induction: After \( 2 \cdot r + 1 \) rounds are all nodes \( \alpha(r + 2)\beta, \beta \in \{0, 1\}^{r+1} \) informed.

- IS: \( r = 0 \) is obvious.

- All nodes \( \alpha(r + 1) \not\in \{\beta_1\}^+, \beta \in \{0, 1\}^{r+1} \) will be informed, because all nodes \( \alpha(r + 2)\beta \) have already received the information.

- If \( \alpha(r + 1) \in \{\beta_1\}^+, \beta \in \{0, 1\}^{r+1} \) holds, then we have \( \alpha(r + 2)\beta a_{r+1} = \alpha(r + 1)\beta_1\beta a_{r+1}. \)

- This node has been informed before.
SE, Proof

\[ \alpha = a_1 a_2 \ldots a_{k-1} a_k \text{ sends to } a_1 a_2 \ldots a_{k-1} \bar{a}_k \text{ (exchange)} ; \]

for \( t = 1 \) to \( k - 1 \) do

    for all \( \beta \in \{0, 1\}^t \) do in parallel begin

        if \( \alpha(t) \not\in \{\beta_1\}^+ \)

            then \( \alpha(t) \beta \text{ sends to } \alpha(t + 1) \beta a_t \text{ (shuffle)} ; \)

            \( \alpha(t + 1) \beta a_t \text{ sends to } \alpha(t + 1) \beta \bar{a}_t \text{ (exchange)} \) end;

Show: All nodes are informed!

- Show by induction: After \( 2 \cdot r + 1 \) rounds are all nodes \( \alpha(r + 2) \beta, \beta \in \{0, 1\}^{r+1} \) informed.

- IS: \( r = 0 \) is obvious.

- All nodes \( \alpha(r + 1) \not\in \{\beta_1\}^+, \beta \in \{0, 1\}^{r+1} \) will be informed, because all nodes \( \alpha(r + 2) \beta \) have already received the information.

- If \( \alpha(r + 1) \in \{\beta_1\}^+, \beta \in \{0, 1\}^{r+1} \) holds, then we have

\[ \alpha(r + 2) \beta a_{r+1} = \alpha(r + 1) \beta_1 \beta a_{r+1} . \]

- This node has been informed before.
Theorem:

We have: \[\left\lfloor \frac{3m}{2} \right\rfloor \leq \text{minb}(BF(m)) = b(BF(m)) \leq 2 \cdot m\]

- The diameter gives the lower bound.
- Algorithm will be provided in the following.
BF

Theorem:
We have: \[ \lfloor \frac{3m}{2} \rfloor \leq \min_b(BF(m)) = b(BF(m)) \leq 2 \cdot m \]

- The diameter gives the lower bound.
- Algorithm will be provided in the following.
Theorem:

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

- 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 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.

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

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

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

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

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

$$\lfloor \frac{3m}{2} \rfloor \leq \text{min}(BF(m)) = b(BF(m)) \leq 2 \cdot m$$
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 $w$. 

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

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

$\left\lceil \frac{3m}{2} \right\rceil \leq \min b(BF(m)) = b(BF(m)) \leq 2 \cdot m$
BF (Proof 1)

- 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 $w$.
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 $w$.

\[
\left\lceil \frac{3m}{2} \right\rceil \leq \min b(BF(m)) = b(BF(m)) \leq 2 \cdot m
\]
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 isomorph.
  - $\#_0(w)$ denotes the number of 0’en in $w$.
  - $\#_1(w)$ denotes the number of 1’en in $w$.

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

- 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.

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

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.
BF (Proof II)

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 $v_0 = (0,00\cdots00)$ 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 $v_1 = (m - 1, 00\cdots01)$ 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 $\lfloor 3m/2 \rfloor$ 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 \( \left\lfloor \frac{3m}{2} \right\rfloor \) 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).
BF (Proof III)

- Aim: Inform in \([3m/2]\) 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).
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:

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

\[
[3m/2] \leq \text{min}b(BF(m)) = b(BF(m)) \leq 2 \cdot m
\]
In $F_0$ a informed node $(l, \alpha 0)$ sends first to $(l + 1, \alpha 0)$ and then to $(l + 1, \alpha(l)0)$. [$\alpha(l) = \alpha_1 \ldots \bar{\alpha}_l \ldots$]

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:

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

- 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 $v_0 = (0, 00 \cdot 00)$ a node $w_0 = (m - 1, \alpha 0)$ is: $1 + \#_0(\alpha) + 2\#_1(\alpha) = m + \#_1(\alpha)$.
- The time to inform from $v_1 = (m - 1, 00 \cdot 01)$ 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: $1 + 2\#_0(\alpha) + \#_1(\alpha) = m + \#_0(\alpha)$. 

\[ \lfloor \frac{3m}{2} \rfloor \leq \min_b(BF(m)) = b(BF(m)) \leq 2 \cdot m \]
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
    \[ m + \#_1(\alpha) < (3m - 1)/2 = \left\lfloor \frac{3m}{2} \right\rfloor. \]
    After this \( w_0 \) sends to \( w_1 \).
    \( w_1 \) is informed at time \( \left\lfloor \frac{3m}{2} \right\rfloor \).
  - **Case 1.2:** \( \#_0(\alpha) < (m - 1)/2 \):
    Node \( w_1 \) will be informed from \( v_0 \) at time
    \[ m + \#_0(\alpha) < (3m - 1)/2 = \left\lfloor \frac{3m}{2} \right\rfloor. \]
    \( w_0 \) will be informed from \( w_1 \) at time \( \left\lfloor \frac{3m}{2} \right\rfloor \).
  - **Case 1.3:** \( \#_0(\alpha) = \#_1(\alpha) = (m - 1)/2 \):
    \( w_0 \) is informed at time
    \[ m + \#_1(\alpha) = (3m - 1)/2 = \left\lfloor \frac{3m}{2} \right\rfloor. \]
    \( w_1 \) is informed at time \( m + \#_0(\alpha) = (3m - 1)/2 = \left\lfloor \frac{3m}{2} \right\rfloor. \)
[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
  $m + \#_1(\alpha) < (3m - 1)/2 = \lfloor 3m/2 \rfloor$.
  
  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
    \[ m + \#_1(\alpha) < (3m - 1)/2 = \lfloor 3m/2 \rfloor. \]
    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
  
  \[ m + \#_1(\alpha) < (3m - 1)/2 = \lfloor 3m/2 \rfloor. \]

  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 $m + #_1(\alpha) < (3m - 1)/2 = \lfloor 3m/2 \rfloor$.
  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 $m + #_0(\alpha) = (3m - 1)/2 = \lfloor 3m/2 \rfloor$. 

\[
\lfloor 3m/2 \rfloor \leq \min b(BF(m)) = b(BF(m)) \leq 2 \cdot m
\]
**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 $m + \#_1(\alpha) \leq 3m/2 - 1 < [3m/2]$. Thus node $w_1$ will be informed at time $[3m/2]$.
  - **Case 2.2:** $\#_0(\alpha) \leq (m - 2)/2$:
    node $w_1$ will be informed from $v_0$ at time $m + \#_0(\alpha) \leq 3m/2 - 1 < [3m/2]$. 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)

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: $\lfloor 3m/2 \rfloor + \lceil m/2 \rceil = 2m$
**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:** $\lfloor 3m/2 \rfloor + \lceil m/2 \rceil = 2m$
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:** $\lfloor 3m/2 \rfloor + \lceil m/2 \rceil = 2m$
Introduction

Broadcast

Complexity

Broadcast on Networks

Lower Bounds

BF (8:47.7)

<>
Walter Unger 6.1.2015 17:11  WS2014/15

BF (Proof V)

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

- **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
    \( m + \#_1(\alpha) \leq 3m/2 - 1 < \lfloor 3m/2 \rfloor \).
    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: \( d \leq \min b(\text{DB}(d)) = b(\text{DB}(d)) \leq \lfloor 3/2 \cdot (d + 1) \rfloor \).

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: \( d \leq \min b(DB(d)) = b(DB(d)) \leq \lfloor 3/2 \cdot (d + 1) \rfloor. \)

Proof:

- Idea \((y_1, y_2, \ldots, y_d)\) informs \((y_2, \ldots, y_d, y_1)\) and \((y_2, \ldots, y_d, \bar{y_1})\).
- 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 \((y_2, \ldots, y_d, \bar{\alpha})\).
- \((0011000)\) informs first \((0110000)\) and then \((0110001)\).
Theorem:
We have: $d \leq \min b(\text{DB}(d)) = b(\text{DB}(d)) \leq \left\lfloor \frac{3}{2} \cdot (d + 1) \right\rfloor$.

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 $(0110001)$. 

![Diagram](image-url)
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: \( d \leq \min b(DB(d)) = b(DB(d)) \leq \lfloor 3/2 \cdot (d + 1) \rfloor. \)

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 \((0110001)\).
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 \((z_1, z_2, \ldots, z_{d-1}, z_d)\).

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

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

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: $\lceil 3(d + 1)/2 \rceil$. 

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 $(z_1, z_2, \ldots, z_{d-1}, z_d)$.

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

- 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 $(z_1, z_2, \ldots, z_{d-1}, z_d)$.

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

- 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: $\lceil 3(d + 1)/2 \rceil$. 
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: $\lceil 3(d + 1)/2 \rceil$. 
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: \( \lceil 3(d + 1)/2 \rceil \).
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 $\text{Broad}_i^A(v_0)$ 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: $|\text{Broad}_i^A(v_0)| = \sum_{s=0}^{i} |\text{Rec}_s^A(v_0)|$. 

---

**Introduction**

Broadcast

Complexity

Broadcast on Networks

Lower Bounds
**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: $|\text{Broad}_i^A(v_0)| = \sum_{s=0}^{i} |\text{Rec}_s^A(v_0)|$. 
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: $|\text{Broad}_i^A(v_0)| = \sum_{s=0}^{i} |\text{Rec}_s^A(v_0)|$. 
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: \( |\text{Broad}_i^A(v_0)| = \sum_{s=0}^{i} |\text{Rec}_s^A(v_0)| \).
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 $t - 2$. 
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 $A(i) = |Rec_i^A(v_0)|$.
- $A(0) = 1$
- $A(1) = 1$
- $A(2) = 2$
- $A(3) = 4$
- $A(i) = A(i - 1) + A(i - 2)$ 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: $A(i) \leq 1.61804^i$ für $i \geq 0$. 
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 $A(i) = |\text{Rec}_i^A(v_0)|$.
- $A(0) = 1$
- $A(1) = 1$
- $A(2) = 2$
- $A(3) = 4$
- $A(i) = A(i - 1) + A(i - 2)$ 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 $A(i) = |\text{Rec}_i^A(v_0)|$.
- $A(0) = 1$
- $A(1) = 1$
- $A(2) = 2$
- $A(3) = 4$
- $A(i) = A(i - 1) + A(i - 2)$ 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

- $A(0) = 1 \leq 1 = 1.61804^0$
- $A(1) = 1 \leq 1.61804 = 1.61804^1$
- $A(2) = 2 \leq 2.61805 = 1.61804^2$
- $A(3) = 4 \leq 4.23612 = 1.61804^3$
- 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: $n \leq |\text{Broad}_t^A(v_0)| = \sum_{i=0}^{t} |\text{Rec}_i^A(v_0)| \leq \sum_{i=0}^{t} A(i) \leq \sum_{i=0}^{t} 1.61804^i = \frac{1.61804^{t+1}-1}{1.61804-1} \leq 3 \cdot 1.61804^t$
- $t \geq 1.4404 \cdot \log_2 n - 3.$

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: $n \leq |Broadcast_t^A(v_0)| = \sum_{i=0}^{t} |Rec_i^A(v_0)| \leq \sum_{i=0}^{t} A(i) \leq \sum_{i=0}^{t} 1.61804^i = \frac{1.61804^{t+1} - 1}{1.61804 - 1} \leq 3 \cdot 1.61804^t$
- $t \geq 1.4404 \cdot \log_2 n - 3$.
- 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: $n \leq |\text{Broadcast}_t^A(v_0)| = \sum_{i=0}^{t} |\text{Rec}_i^A(v_0)| \leq \sum_{i=0}^{t} A(i) \leq \sum_{i=0}^{t} 1.61804^i = \frac{1.61804^{t+1} - 1}{1.61804 - 1} \leq 3 \cdot 1.61804^t$

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

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

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

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

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

Proof of the second statement my 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: $n \leq |\text{Broadcast}^A(v_0)| = \sum_{i=0}^t |\text{Rec}^A_i(v_0)| \leq \sum_{i=0}^t A(i) \leq \sum_{i=0}^t 1.61804^i = \frac{1.61804^{t+1} - 1}{1.61804 - 1} \leq 3 \cdot 1.61804^t$

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

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.
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:
\[ b(DB_k) \geq \text{minb}(DB_k) \geq 1.1374 \cdot k - 2 \]

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

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 |
|---------|------|----------|-------------|-------------|
| $K_n$   | $n$  | $1$      | $\lceil \log_2 n \rceil$ | $\lceil \log_2 n \rceil$ |
| $HQ_k$  | $2^k$| $k$      | $k$         | $k$         |
| $CCC_k$ | $k \cdot 2^k$ | $\lceil 5k/2 \rceil - 2$ | $\lceil 5k/2 \rceil - 2$ | $\lceil 5k/2 \rceil - 2$ |
| $SE_k$  | $2^k$| $2k - 1$ | $2k - 1$    | $2k - 1$    |
| $DB_k$  | $2^k$| $k$      | $1.4404k$   | $\frac{3}{2}(k + 1)$ |
| $BF_k$  | $k \cdot 2^k$ | $\lceil 3k/2 \rceil$ | $1.7609k$   | $2k - \frac{1}{2} \log \log k + c$ |
J. Hromkovič, et al.:
Questions

- Give the idea for the NP-completeness proof for the broadcast problem?
- Give the idea for the broadcast on the following networks
  - CCC
  - BF
  - SE
  - DB
- What are the ideas for the lower bounds for the broadcast problem?
Questions

- Give the idea for the NP-completeness proof for the broadcast problem?

- Give the idea for the broadcast on the following networks
  - CCC
  - BF
  - SE
  - DB

- What are the ideas for the lower bounds for the broadcast problem?
Questions

- Give the idea for the NP-completeness proof for the broadcast problem?
- Give the idea for the broadcast on the following networks
  - CCC
  - BF
  - SE
  - DB
- What are the ideas for the lower bounds for the broadcast problem?
Questions

- Give the idea for the NP-completeness proof for the broadcast problem?
- Give the idea for the broadcast on the following networks
  - CCC
  - BF
  - SE
  - DB
- What are the ideas for the lower bounds for the broadcast problem?
Questions

- Give the idea for the NP-completeness proof for the broadcast problem?

- Give the idea for the broadcast on the following networks
  - CCC
  - BF
  - SE
  - DB

- What are the ideas for the lower bounds for the broadcast problem?
Questions

- Give the idea for the NP-completeness proof for the broadcast problem?
- Give the idea for the broadcast on the following networks
  - CCC
  - BF
  - SE
  - DB
- What are the ideas for the lower bounds for the broadcast problem?
Questions

- Give the idea for the NP-completeness proof for the broadcast problem?
- Give the idea for the broadcast on the following networks
  - CCC
  - BF
  - SE
  - DB
- What are the ideas for the lower bounds for the broadcast problem?
Questions

- Give the idea for the NP-completeness proof for the broadcast problem?
- Give the idea for the broadcast on the following networks
  - CCC
  - BF
  - SE
  - DB
- What are the ideas for the lower bounds for the broadcast problem?
<table>
<thead>
<tr>
<th>Legend</th>
</tr>
</thead>
<tbody>
<tr>
<td>■  : Not of relevance</td>
</tr>
<tr>
<td>▶  : implicitly used basics</td>
</tr>
<tr>
<td>▶  : idea of proof or algorithm</td>
</tr>
<tr>
<td>▶  : structure of proof or algorithm</td>
</tr>
<tr>
<td>▶  : Full knowledge</td>
</tr>
</tbody>
</table>