ONLINE ALGORITHMS
Exercise Sheet 3

PD Dr. Walter Unger
Janosch Fuchs
Department of Computer Science
RWTH Aachen University

SS 17
May 12
Due Date: May 19, 10:00
(Office 4014 at I1)

• Exercises appear at the i1 homepage (http://algo.rwth-aachen.de/en/Lehre/SS17/Online.php) on Friday.
• You have seven days to create a solution and it must be done in a group of two or three students.
• Write the name, group number and enrollment number of each group member on every sheet that you hand in.
• To achieve the permission for the exam you must earn 50% of the sum of all points and present one of your solutions at least once.
• You can earn 50% bonus points by presenting your solution. At the beginning of every exercise session, you can mark the exercises that you want to present.
• If a student is not able to present a correct solution although he/she marked the exercise as presentable, he/she will lose all of his/her points on the exercise sheet.

Exercise 1

(4 points)

As the cow arrives in the barn, she meets an old friend that tells her about a funny game. The game is called the zero-sum game. The game is played on a matrix \( A \) with arbitrary dimension. Because the friend of the cow is very friendly, the cow is allowed to start. She can choose one column \( j \) of the matrix. Then, the other player can choose a row \( i \). The cow wins \( a_{i,j} \) points and her friend \(-a_{i,j}\). Each player wants to maximize their points.

\[
A_1 = \begin{bmatrix}
4 & 2 & 0 & 1 \\
-1 & 3 & 1 & 4 \\
1 & -4 & -1 & 15 \\
10 & 22 & 12 & 0
\end{bmatrix}
\]

a) What would you do if you would play against the friend of the cow? Which column of \( A_1 \) would you choose? Explain why.

b) As the cow, she does the obvious: She mentions that \( \forall A = (m_{i,j}) : \max_j \min_i m_{i,j} \leq \min_i \max_j m_{i,j} \). Her friend looks a little bit confused. Could you proof this for him?

c) The friend of the cow seems not convinced. Give him an example such that \( \max_j \min_i m_{i,j} < \min_i \max_j m_{i,j} \) holds.

The cow suggests that the game gets more interesting if one player would choose a probability distribution with which the columns/rows are played. The choice from the probabilistic distribution is made at the end. So, the player that is allowed to create a distribution starts, reveals its distribution and then the other player can choose the row/column.

d) What is the estimated gain/cost if the cow would choose the first column with a probability of \( \frac{1}{2} \) and the others with \( \frac{1}{6} \)?

e) What is the estimated gain/cost if the other player chooses the first and last row with a probability of \( \frac{1}{4} \) and the second and third with \( \frac{1}{6} \)?

Exercise 2

(3 points)

Assume you have two machines \( m_1 \) and \( m_2 \) that work on tasks \( t_i \in \mathbb{N} \), with \( i \leq n \). A task \( t_i \) needs exactly \( t_i \) time steps on one machine to be completed and arrives in time step \( i \). You cannot split tasks and your machines cannot work in parallel. You have to assign each task on one machine. You need to minimize the largest sum of the tasks of one machine. So, your task is to minimize \( \max_j (\sum_{t_i \in m_j} t_i) \).

a) Prove a strict lower bound on the competitive ratio of \( \frac{3}{2} \).
b) Assume you can make use of two random bits. Show that this can help for your worst case instance from a). What is the new estimated finish time?

c) Derive a lower bound for the competitive ratio of the following randomized algorithm. Each time a task arrives, it flips a coin. If it shows heads, $m_1$ is chosen and $m_2$ otherwise.

**Exercise 3** (8 points)

Graph exploration: The graph exploration problem in a known undirected graph structure with unknown edge weights, denoted by GEP-UW, is the following minimization problem. An undirected graph $G = (V, E)$ with a start vertex $s_0 \in V$ is given. An agent is located in $s_0$, and its goal is to find a path through the graph visiting each vertex at least once. In one time step, the agent can move from its current vertex to any of the neighboring vertices. Whenever the agent visits a vertex for the first time, the weights of all incident edges are revealed. The agent knows the graph structure from the beginning and has unlimited computing power, in particular, it can remember all edge weights seen so far. The goal is to minimize the sum of edge weights on the path traveled by the agent. More precisely, for some algorithm $ALG$ using the agent on some graph $G$, we denote by $p_A = v_1, \ldots, v_\ell$ the path traveled by the agent, and by $COST_{ALG}(G) = \sum_{i=1}^{\ell-1} c(v_i, v_{i+1})$ the cost of $p_A$.

Grid Graph: For $m, n \geq 1$, the undirected $m \times n$ grid is the graph $G_{m,n} = (V, E)$, where $V = \{(i, j) \mid 1 \leq i \leq m, 1 \leq j \leq n\}$ and $E = \{(i, j), (i + 1, j) \mid 1 \leq i \leq m - 1, 1 \leq j \leq n\} \cup \{(i, j), (i, j + 1) \mid 1 \leq i \leq m, 1 \leq j \leq n - 1\}$.

a) Derive a lower bound of at least $\frac{5}{4}$ for the GEP-UW with only two weights 1 and $k > 1$ on Grids with $m = 2$.

b) Derive an upper bound of at most 3 for the GEP-UW with only two weights 1 and $k > 1$ on Grids with $m = 2$. Give a detailed idea.