Prof. Dr. Susanne Albers
Dr. Suzanne van der Ster
Dario Frascaria
Lehrstuhl für Theoretische Informatik
Fakultät für Informatik
Technische Universität München

Problem set 7
May 27, 2016
Summer Semester 2016

Online and Approximation Algorithms

Due June 3, 2016 before 10:00

Exercise 1 (Path Game - 10 points)

Consider the following 2-player game. There is a graph $G = (V, E)$ and the game takes place in alternating turns. In each turn, a player picks an edge $e \in E$ which has not been chosen by any player before, so that the selected edges form a single path. The first player who is unable to choose such an edge loses the game.
Show that, if the starting player is given a perfect matching M of G, there exists a winning strategy for him.

Exercise 2 (Randomized Matching - 10 points)

Consider the following randomized online algorithm for the maximum matching problem on bipartite graphs. Whenever a new vertex $v \in V$ arrives, match v with a vertex $u \in U$ chosen uniformly at random among the currently unmatched neighbors of v. Show that the competitive ratio of this algorithm cannot be better than $\frac{1}{2}$.

Hint: Consider a bipartite graph $G = (U \cup V, E)$ such that $U = \{u_1, u_2, \ldots, u_n\}$ and $V = \{v_1, v_2, \ldots, v_n\}$. The vertices u_i and v_j are connected if and only if either $1 \leq i, j \leq \frac{n}{2}$, or $i + j = n + 1$.

Exercise 3 (Ranking - 10 points)

In the lecture it was shown that, when analyzing the *Ranking* algorithm, one can focus on graphs with a perfect matching. Given a bipartite graph $G = (U \cup V, E)$, it was proved that the removal of vertices in U can only decrease the size of the matching produced by Ranking.
Now prove the same for the removal of vertices from V.

Exercise 4 (Ranking II - 10 points)

Let $G = (U \cup V, E)$ be a bipartite graph. Prove that the *Ranking* algorithm fulfills the following property.
When fixing a permutation π on U, the following methods produce the same matching:

Method 1 Nodes of V arrive online and each node $v \in V$ is matched to an adjacent node $u \in U$ that has the lowest rank according to π.

Method 2 Nodes in $V = \{v_1, ..., v_{|V|}\}$ are known in advance and nodes in U arrive in an online fashion according to π. Every node $u \in U$ is matched to an adjacent node $v \in V$ with the lowest index number.