Abteilung Algorithmik
Institut für Betriebssysteme und Rechnerverbund **SS 11**
TU Braunschweig

Prof. Dr. Sándor Fekete
Dr. Iris Reinbacher

Online-Algorithms
2nd Homework Assignment, 2. May 2011

Due on 16. May 2011 until 13:00 in the box opposite IZ 252 or in PK 11.1
Don't forget to label each sheet with your name!

Exercise 1 (Randomized Algorithms):

a) For some problem P, two online algorithms A_1 and A_2 are given, with a competitve ratio of 2 and 3, respectively. Design a randomized online-algorithm with competitive ratio 9/4.

b) During the last lab we presented the RSUM algorithm for the Bahncard Problem. What is the competitve factor of this algorithm for the real world Bahncard Problem?

(5 + 5 points)

Exercise 2 (k-Server Problem):
Consider the k-server problem: The algorithm can move k servers in space; in the beginning they are placed on fixed points of a set M in space.
Given is a sequence of requests $\sigma = r_1, r_2, \ldots, r_n$, where each request corresponds to a point in the plane. A request r_i is considered *served*, when a server has reached the point r_i.
The algorithm has to serve the requests in the given order by moving the servers. The cost of the algorithm is the sum of all distances that the servers have to move (according to some specified metric).

Show that a greedy strategy for the k-server problem is not necessarily competitive. (Here, a greedy strategy chooses the cheapest possibility, i.e., it moves the server that is closest to the request.)

Hint: Consider an example with $k = 2$ servers and an infinite sequence on 3 well-chosen request points.

(20 points)

Exercise 3 (Memory):
We consider a version of the game "Memory" for one player. As in the usual game, let n pairs of cards lie on the table, face down. In each move of the game, the player can turn two cards. If they are identical, they are removed, otherwise they are turned face down again and put back in their spots. The goal of the player is to remove all cards with the least number of moves. Even after the successful turning of a pair, any further turning is considered a move.
We assume that the player can remember all cards that have been turned at any time. The optimal offline strategy needs n moves to remove all $2n$ cards.

a) Design an online strategy that takes at most $2n$ moves, i.e., that is 2-competitive.

b) Design an online strategy that takes at most $2n - 1$ moves.

c) Show that any deterministic strategy can be forced to take at least $2n - 1$ moves.

d) What changes if the player gets a free move after removing a pair?

(5 + 10 + 10 + 5 points)