Abteilung Algorithmik
Institut für Betriebssysteme und Rechnerverbund **SS 13**
TU Braunschweig

Prof. Dr. Sándor Fekete
Dr. Christiane Schmidt
Dr. Michael Hemmer

Online-Algorithms
0th Homework Assignment, 16. April 2013

You do not need to hand in this assignment, it will be treated in the first small tutorial on April 30

Exercise 1 (Ski Rental Problem):
Let R be the ratio of the purchase price p and the lending fee ℓ ($R = \frac{p}{\ell}$).

a) Design a strategy that reaches a competitive ratio of $2 - \frac{1}{R}$. Prove the correctness of your strategy.

b) In the following, let $p = 100, \ell = 20$.

i) What is the competitive ratio of the strategy *rent for 3 days, buy on the 4th*?

ii) What is the competitive ratio of the optimal strategy?

iii) Design a strategy with competitive ratio 3.

Exercise 2 (Bin Packing):
For the problem BIN PACKING you are given n objects of size $s_1, \ldots s_n \in [0, 1]$ and an unlimited number of bins with capacity 1 each. The objective is to use as few bins as possible.
More formally, we are looking for the smalles number k for which the set $\{1, \ldots, n\}$ can be partitioned into k sets $B_1, \ldots, B_k$ such that for each $i \in \{1, \ldots, n\}$ the equation $\sum_{j \in B_i} s_j \leq 1$ is satisfied.
In the online version of BIN PACKING every object j has to be assigned to a bin without knowing the number n and without knowing the sizes of the objects $(j + 1), \ldots, n$.
Give a 2-competitve online algorithm for BIN PACKING and prove its competitive ratio.

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 over two cards. If they are identical, they are removed; otherwise they are turned over

again and put back into their spots. The goal of the player is to remove all cards with the least number of moves. Even after successfully identifying 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?