Abteilung Algorithmik **Summer term 2016**
Institut für Betriebssysteme und Rechnerverbund
TU Braunschweig

Prof. Dr. Sándor P. Fekete
Phillip Keldenich

Online Algorithms
0^{th} Homework Assignment, 18^{th} of April, 2016

You **do not** have to hand in this exercise sheet. The solutions will be presented in the small tutorial on Wednesday, the 27^{th} of April 2016.

Exercise 1 (Memory): We consider the well-known game MEMORY in a single-player variant. There are n pairs of cards in a row that the player has to remove by uncovering matching pairs in the fewest moves possible.

In each move, the player uncovers a first card and a second card. He *can* see the first card before choosing the second card. If the cards match, both are removed. Every move, even a move removing a pair of cards, incurs a cost of 1; thus, the optimal offline algorithm requires exactly n moves to remove all cards.

a) Design a 2-competitive online algorithm for MEMORY, i.e. a strategy that requires at most $2n$ moves.

b) Design a $\frac{3}{2}$-competitive online algorithm for MEMORY.

c) Is there a c-competitive online algorithm if the cost of a move that removes a pair is set to 0 (mirroring the rules of the original 2-player game where a player can keep playing after uncovering a pair?)

(0 points)

Exercise 2 (Potential Functions): In the analysis of online algorithms it is often impossible to bound the cost of the online algorithm in terms of the optimal cost independently for every single request. Instead, one has to find a way to distribute the cost of an expensive operation the online algorithm performs across several requests.

One way to do this is by using a potential function $\Phi_\sigma : \{0, 1, \ldots, n\} \to \mathbb{R}_{\geq 0}$, where $\Phi_\sigma(i)$ denotes the value of the potential function after request i and $\Phi_\sigma(0) = 0$. In some way, Φ_σ is a *savings account* for cost which is not allowed to run negative and which can be used to store earlier savings to finance more expensive operations later on.

a) For an online algorithm A, let $c_A(\sigma)$ be the cost of the algorithm on the entire sequence and $c_A(i)$ be the cost on request i (and similarly for OPT, the optimal offline algorithm). Prove the following: If for every request sequence σ there exists a potential function Φ_σ such that for all $1 \leq i \leq n$ we have

$$c_A(i) + \Phi_\sigma(i) - \Phi_\sigma(i-1) \leq c \cdot c_{OPT}(i),$$

then A is c-competitive, i.e. $c_A(\sigma) \leq c \cdot c_{OPT}(\sigma)$ for all request sequences σ.

b) Consider the online problem READ INTO ARRAY: The task is to read an unknown-length non-empty string $s \in \Sigma^+$ of symbols into a list stored as contiguous array one symbol at a time. The array may be over-allocated, subject the the restriction that the the array may never be larger than $2|s|$. Reading a symbol has a basic cost of 1. When the currently allocated array is full, a new, larger array has to be allocated and the contents must be copied, incurring a cost of 1 for each symbol already stored in the array. In other words, the cost for the kth symbol is either 1 (not full) or k (full).

Use a potential function to show that the algorithm DOUBLE SIZE that starts with an array of size 1 and doubles the array size each time the array is full has a competitive ratio of 3.

(0 points)