Abteilung Algorithmik **Summer term 2018**
Institut für Betriebssysteme und Rechnerverbund
TU Braunschweig

Prof. Dr. Sándor P. Fekete
Phillip Keldenich

Online Algorithms
0th Homework Assignment, 11th of April, 2018

You **do not** have to hand in this exercise sheet. The solutions will be presented in the small tutorial on Monday, the 23rd of April 2018.

Exercise 1 (Memory): We consider a single-player version of the well-known game MEMORY. There are n pairs of cards lying face down in a row. 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. Clearly, the optimal offline algorithm that knows the face of all cards 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) Prove that there cannot be an online algorithm for MEMORY that uses less than $2n - 1$ moves in the worst case.

c) In the original 2-player game, a player may perform another move after he uncovers a matching pair. Is there a c-competitive online algorithm if the cost of a move that removes a matching pair is set to 0?

(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. $c_A(\sigma) \leq c \cdot c_{OPT}(\sigma)$.

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

Devise an online algorithm for this problem and use a potential function to prove that its competitive ratio is at most 3.

(0 points)