Prof. Dr. Susanne Albers
Dr. Dimitrios Letsios
Dario Frascaria
Lehrstuhl für Theoretische Informatik
Fakultät für Informatik
Technische Universität München

Problem set 2
April 27, 2015
Summer Semester 2015

# Online and Approximation Algorithms

*Due May 4, 2015 before class!*

## Exercise 1 (LIFO and LFU- 10 points)

LIFO (Last In-First Out) is the online paging algorithm that evicts the page most recently moved to the fast memory and LFU (Least Frequently Used) is the online paging algorithm that replaces the page that has been used least since it entered the cache. Prove that LIFO and LFU are not $c$-competitive for any constant $c > 0$.

## Exercise 2 (Marking Algorithm - 10 points)

Consider a sequence of requests $\sigma$ which request pages in a memory system with cache size $k$. A *$k$-phase partition* of $\sigma$ is obtained as follows: we partition the request sequence into phases such that each phase is the maximal sequence containing $k$ pairwise distinct pages that follows the previous one, except possibly the last phase which contains requests to at most $k$ different pages.
Given a $k$-phase partition of $\sigma$, we define a marking of the pages requested as follows. At the beginning of a phase, all pages are unmarked. During phase $i$, a page is marked upon the first request to it. Recall that an online paging algorithm is a *marking algorithm*, if it never evicts a marked page.

1. Prove that every marking algorithm is $k$-competitive.

2. Prove that FIFO is not a marking algorithm.

## Exercise 3 (Amortized Analysis, Sequence of Requests - 10 points)

Suppose that we serve a sequence of $n$ requests, where the cost for serving $i$-th request is

$$c_i = \begin{cases} i, & \text{if } i \text{ is an exact power of 2} \\ 1, & \text{otherwise} \end{cases}$$

What is the worst-case cost for serving the requests? Use amortized analysis with an appropriate potential function in order to show that the total cost is upper bounded by $3n$.

## Exercise 4 (Amortized Analysis, Blocks into a Box - 10 points)

Consider an initially empty box on which we can store at most $n$ blocks in the form of a tower. We perform a sequence of $n$ operations, where each operation is one of the following:

1. Put a block at the top of the tower with cost 1.
2. Remove the block from the top of the tower with cost 1.
3. Remove the $k$ top blocks from the tower with cost $k$.

What is the worst-case cost for serving the requests? Use amortized analysis with an appropriate potential function in order to show that the total cost is upper bounded by $2n$.