Prof. Dr. Susanne Albers
Dr. Suzanne van der Ster
Dario Frascaria
Lehrstuhl für Theoretische Informatik
Fakultät für Informatik
Technische Universität München

Problem set 2
April 22, 2016
Summer Semester 2016

Online and Approximation Algorithms

Due April 29, 2016 before 10:00

Exercise 1 (Marking Algorithm - 10 points)

Consider a sequence of requests σ for pages from a memory system with a fast memory of size k. A *k-phase partition* of σ 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 σ, we define a *marking* of the pages requested as follows. At the beginning of a phase, all pages are unmarked. During the phase, 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.

(a) Prove that every marking algorithm is k-competitive.

(b) Prove that FIFO is not a marking algorithm.

Exercise 2 (Conservative Algorithm - 10 points)

Consider a sequence of requests σ for pages from a memory system with a fast memory of size k. We say that a paging algorithm is *conservative* if on any consecutive input subsequence containing k or fewer distinct page references, the algorithm will incur k or fewer page faults.
Recall that the algorithm Flush When Full (FWF), upon a page fault, when there is no space left in fast memory, evicts all pages currently in fast memory.
Show that FWF is not a conservative algorithm.

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

Suppose that we serve a sequence of n requests, where the cost for serving the ith request is

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

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

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

Consider an initially empty box in 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.

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