Technische Universität München
Fakultät für Informatik
Lehrstuhl für Theoretische Informatik
Prof. Dr. Susanne Albers
Jens Quedenfeld

Winter Semester 2017/18
Problem Set 5
November 20, 2017

Online and Approximation Algorithms

Due November 27, 2017 at 10:00

Exercise 1 (Dynamic List Update problem – 10 points)

We modify the list update problem by adding the operations INSERT(a) and DELETE(a). Let n be the current length of the list. Whenever INSERT(a) is called, the list is searched for element a. If a is already in the list, the cost of the operation is equal to a's position. If not, it is inserted at position $n+1$ of the list. In this case, the cost is $n+1$. The algorithm can move the inserted item at no cost to any position closer to the front of the list.
If DELETE(a) is called, the list is searched for element a. If a is found, the cost for the operation is equal to a's position, otherwise the cost is n.
The algorithm Move-To-Front (MTF) moves the newly inserted element to the front of the list. Show that MTF remains 2-competitive.

Exercise 2 (String Scanning, Yao's Principle – 10 points)

Consider the following problem. Given a string $x \in \{0,1\}^n$, we want to determine if x contains two consecutive 1. By using Yao's Minimax Principle, show that the expected number of bits inspected by any randomized algorithm is $\Omega(n)$.

Exercise 3 (Move-To-Front-2 – 10 points)

The algorithm MTF2 for the list update problem works as follows. In contrast to MTF an element is moved to the front after it is accessed the second time. For each element x there is a bit $b(x)$ that is flipped after each access. If the bit is flipped from 0 to 1, the element is moved to the front. Initially, all bits are 1.
Show that MTF2 is 3-competitive. Consider the static model (i.e. without the operations INSERT(a) and DELETE(a)).
Hint: Use the term $b(x)$ in the potential function.

Exercise 4 (Modified BIT – 10 points)

Recall the BIT algorithm for the list update problem which assigns a random bit to every item before any request is served. When an item is requested, then its bit is flipped. If the bit becomes 1, then the item is moved to the front of the list. Otherwise, its position does not change.
We modify the algorithm as follows. If the requested item is already in front of the list, then we do not flip its bit. Show that the modified algorithm is no longer $\frac{7}{4}$-competitive.