Universität
des
Saarlandes

Kurt Mehlhorn, Konstantinos Panagiotou and Reto Spöhel

WS 2010-11

Models of Computation, an Algorithmic Perspective

Assignment 8 Tue 07.12.2010

This assignment is **due on December 15/17** in your respective tutorial groups. You are allowed (even encouraged) to discuss these problems with your fellow classmates. All submitted work, however, must be *written individually* without consulting someone else's solutions or any other source like the web.

Exercise 1 Consider the pointer representation of a tree T with n vertices. Each vertex points to its parent with the root pointing to '*nil*'. The pointers are stored in a pointer array $P[1 \dots n]$. Additionally each vertex is associated with a value which is stored in the array $A[1 \dots n]$.

Now for each vertex i we want to compute the largest value appearing on the path from vertex i to the root. Design a deterministic algorithm that solves the above problem with depth $O(\log n)$ and work $O(n \log n)$.

Exercise 2 Consider the randomized algorithm for computing prefix sums in a list of size n that was sketched in the lecture.

a) Work out the implementation details, i.e., give the pseudocode and argue about its correctness.

b) Show that the algorithm has depth $O(\log^2 n)$ with high probability. [Hint: Bound the recursion depth by computing the expected number of remaining list elements at recursion depth i and applying Markov's inequality.]

c) Modify the algorithm as hinted at in the lecture to achieve a depth of $O(\log n \log \log n)$ w.h.p.

d) Show that the (improved) algorithm has work $O(n)$ w.h.p. [Hint: Use e.g. Chebyshev's inequality to bound the probability that in a given recursion step, less than 1/8 of the remaining list elements are removed.]