TEL AVIV UNIVERSITY
Department of Computer Science
0368.4281 – Advanced topics in data structures
Spring Semester, 2007/2008

Homework 1, May 28, 2008

Due on June 11. Don't forget to keep a copy of the homework.

1. Consider a union-find data structure with union by rank but no path compression on a set of n elements (we start by n make-sets and then continue only with unions and finds).
(a) Prove that every node has rank at most $\lfloor \log(n) \rfloor$.
(b) Prove that any sequence of m operations takes $O(m \log n)$ time.
(c) For each $m > 2n$ show a sequence of m operations that takes $\geq cm \log n$ time (not counting the make-sets) where c is some constant.

2. Show a sequence of operations on a union find data structure with union by rank and path compression such that in the resulting forest there is a node x with $level(p(x)) < level(x)$. Here x is not a root and $p(x)$ is the parent of x.

3. Consider Ackermann's function defined for $n = 1, 2, 3, ...$ as follows $A_1(n) = 2n$, and $A_k(n) = A_{k-1}^n(1)$. Let $\alpha_r(n)$ (for $n \geq 2$) be the inverse of A_r that is $\alpha_r(n) = \min\{k | A_r(k) \geq n\}$.
(a) Prove that for $r \geq 3$ $\alpha_r(n^2) \leq 1 + \alpha_r(n)$.
Now let $\hat{\alpha}_2(n) = \alpha_2(n)$, and for $r > 2$, $\hat{\alpha}_r(n) = 0$ if $n < 16$, and $\hat{\alpha}_r(n) = 1 + \hat{\alpha}_r(2\hat{\alpha}_{r-1}(n))$ if $n \geq 16$.
(b) Write a function $\hat{A}_r(n)$ such that $\hat{\alpha}_r(n) = \min\{k | \hat{A}_r(k) \geq n\}$ for every $r \geq 2$, and $n \geq 16$.
(c) Prove that $\hat{\alpha}_r(n) = O(\alpha_r(n))$.

4. (a) Show how to preprocess a list of n elements from a semigroup so that your data structure takes $O(n)$ space and you answer an interval product query in $O(\alpha(n))$ time.
(b) Show how to preprocess a list of n elements from a group so that your data structure takes $O(n)$ space and you answer a interval product query in $O(1)$ time.

5. Consider the following game. I choose an integer n and keep it secret. Your goal is to discover n. We play the game in rounds, where in each round you send a list of integers $y_1, \ldots, y_m$ such that $m \leq n$ (otherwise you loose right away). Then I have to answer and say for each y_j whether $y_j > n$, $y_j < n$, or $y_j - n$. Show how you can always discover n in $O(\alpha(n))$ rounds.