TEL AVIV UNIVERSITY
Department of Computer Science
0368.4281 – Advanced topics in data structures
Spring Semester, 2000/2001

Homework 4, June 6, 2001

Due on Sunday June 24 in my mailbox.

1. We say that a flow f is acyclic if there is no cycle with positive flow along it. Given a flow f from s to t in a directed graph $G = (V, E)$ show how to get from f an acyclic flow f' such that $|f| = |f'|$ (try to get an algorithm that runs in $O(m \log n)$ time.).

2. Extend the data structure of Dietz and Sleator for the "order maintenance" problem described in class to a more efficient data structure in which every operation costs $O(1)$ amortized time.

3. We build a tree dynamically using the operation $insert(x, y)$ that adds node x as a child of node y which is already in the tree. At any time between the insert operations we would like to be able to answer queries of the form $ancestor(x, y)$: Is x an ancestor of y in the tree ? Describe an efficient algorithm for the problem.

4. Extend the fully persistent queues presented in class such that they support the following operations.
1) $Q' = inject(Q, x, d)$ that produce Q' by inserting element x immediately before the d last elements in the queue Q.
2) $(x, Q') = pop(Q, d)$ that produce a queue Q' from Q by taking out the dth element, x, from the begining of Q.
What are the time and space complexities of your data structure ?

5. Describe an implementation of fully persistent heaps that support the operations *find-min*, *insert*, and *delete-min*. Make your implementation as efficient as you can. What are the time and space complexities of your data structure ?