TEL AVIV UNIVERSITY
Department of Computer Science
0368.4281 – Advanced topics in data structures
Spring Semester, 2005/2006

Homework 4, Jan 4, 2006

Due on Thu. Jan 19.

1. Show how to augment dynamic trees to support an operation $sumcost(v)$ that returns the sum of the costs of the edges on the path from v to the root of the tree containing it. An efficient implementation should run in $O(\log n)$ time. Do not hurt the logarithmic running time of any of the other operations.

2. Show how to extend dynamic trees to support an operation called $evert(v)$. This operation changes the root of the (actual) tree containing v to be v. In other words we change the direction of the edges on the path from v to the previous root of the tree. An efficient implementation should run in $O(\log n)$ time. Do not hurt the logarithmic running time of any of the other operations.

3. 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.).

4. On a set of n nodes we perform a sequence of operations, each of which is one of the followings.
(a) $insert(u, v)$: Adds an edge between u and v.
(b) $delete - oldest$: Removes the edge that was inserted first among the edges currently in the graph.
(c) $connected(u, v)$: Answers true if there is a path from u to v in the current graph. Otherwise answers false.

Describe a data structure that supports these operations and analyze its performance. (A structure supporting each operation in $O(\log n)$ amortized time would receive maximum score.)