TEL AVIV UNIVERSITY
Department of Computer Science
0368.4281 – Advanced topics in data structures
Fall Semester, 2004/2005

Homework 4, November 21, 2004

Due on Monday December 6. Don't forget to keep a copy of the homework.

1. Consider the move to root heuristic on a tree containing the integers 1 to n in symmetric order (i.e. for every node v all items in the left subtree are smaller than the item at v, and the item at v is smaller than all items at the right subtree). Recall that in the move to root heuristic, after you access an item you perform rotations bottom up on the edge between the accessed item to its parent until the item is at the root. Assume $i < j$.
a) Prove that i is an ancestor of j if the most recent request for i has occured since the most recent request for any of $i+1, \ldots, j$. Similarly, j is an ancestor of i if the most recent request for j has occured since the most recent one for any of $i, \ldots, j-1$.
b) Prove that any tree T containing these items in symmetric order can be produced from any other tree containing these items in symmetric order by a sequence of at most n requests under the move to root heuristics.
c) For any m sufficiently large show a sequence of accesses such that serving it with the move to root heuristics takes $\Omega(n)$ time on average per access. (The initial tree is arbitrary.)

2. We define the following variation on the splay algorithm. This variation looks 3 steps (edges) towards the root from the node x and applies one of the rules in Figure 1 (or their mirror image) if possible. If it is not possible to apply one of the rules in Figure 1 we apply one of the regular zig-zig, zig-zag, or zig rules (Note that zig or zig-zig would apply only if x is at distance 1 or 2 from the root, respectively).

Prove that the access lemma holds for this variation as well (with a different constant).

3. Assume that the dynamic optimality is true and prove that to splay the nodes of a binary tree one by one in symmetric order takes $O(n)$ time where n is the number of nodes in the tree.

4. Consider the generic preflow push algorithm where the next active vertex we choose is always an active node v with maximum distance label (d(v)).
a) Prove that this algorithm performs at most $O(n^3)$ nonsaturating pushes.
b) Suggest an implementation of this algorithm that runs in $O(n^3)$ time.

5. 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.).

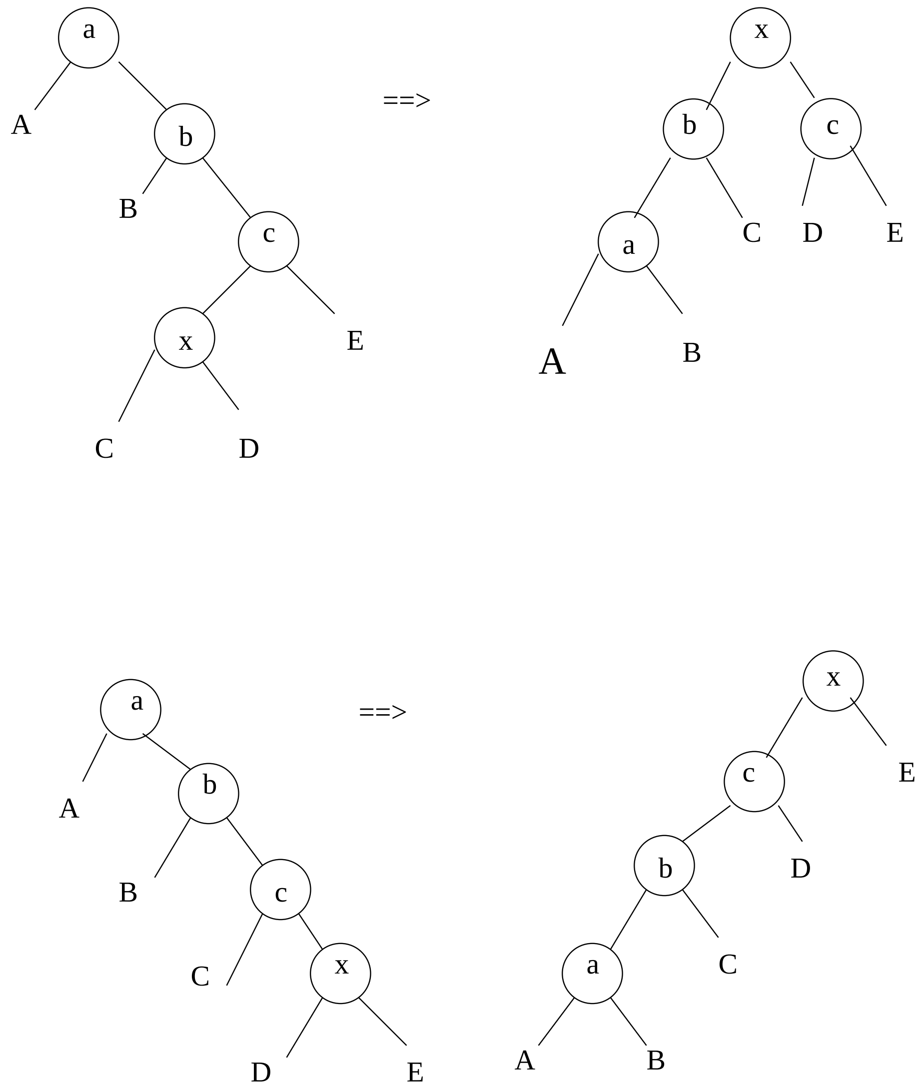

Figure 1: Splay cases in question (3)