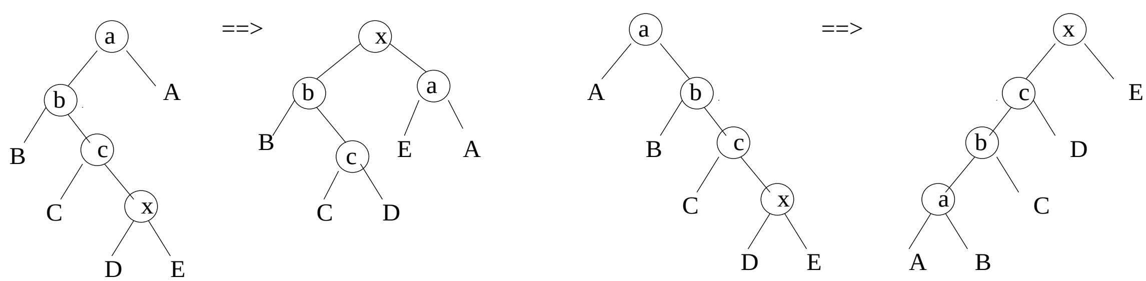
TEL AVIV UNIVERSITY
Department of Computer Science
0368.4281 – Advanced topics in data structures
Spring Semester, 2002/2003

Homework 4, May 16, 2003

Due on Mon. June 2

1. Let $x_1, \ldots, x_n$ be a sequence of distinct numbers. An inversion is a pair i, j where $i < j$ and $x_i > x_j$. Describe an algorithm that sorts a sequence $x_1, \ldots, x_n$ in $O(n \log(\frac{I}{n} + 2))$ time, where I is the total number of inversions in the sequence $x_1, \ldots, x_n$.

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).



(1) (2)

Figure 1: Splay cases in question (3)

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. Show how to extend dynamic trees to support an operation called $evert(v)$. This operation should change the root of the tree containing v to be v.

5. 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.)