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

Homework 4, December 27, 2004

Due on Monday January 10. Don't forget to keep a copy of the homework.

1. Describe how to implement the tries of Willard presented in class, so that an insertion or a deletion of the largest or the smallest item in the list takes $O(1)$ amortized time (without hurting the time bound of the other operations). Argue that your implementation indeed achieves the desired bounds.

2. Extend the data structure described in class for the "order maintenance" problem 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. Use persistent search trees to solve the following problem. Given a set S of n points in the plane, preprocess them to build a data structures which given a triple x_1, x_2, y_1 allows to report efficiently all the points $(x, y) \in S$ such that $x_1 \leq x \leq x_2$ and $y \geq y_1$. Analyze the space, preprocessing time, and query time 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 ?