TEL AVIV UNIVERSITY
Department of Computer Science
0368.4281 – Advanced topics in data structures
Spring Semester, 2003/2004

Homework 5, June 8, 2004

Due on Friday June 26. Don't forget to keep a copy of the homework.

1. 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.

2. Use persistent data structures to describe an efficient and simple solution to the following problems. Analyze the space, preprocessing time, and query time required by your solution.

   1. Given a set S, of n intervals $\{[a_1, b_1], [a_2, b_2], \ldots, [a_n, b_n]\}$ on the real line, build a data structure that supports an efficient query of the form: "Given a point x find all intervals that contain x".

   2. Given a set S of n segments in the plane all parallel to the x-axis, build a data structure that supports an efficient query of the form: "Given a segment s parallel to the y-axis find all segments in S that intersect s".

3. 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 ?