CSE 332 Data Abstractions, Winter 2012

Homework 3

Due Friday, Jan 27, 2012 at the beginning of lecture. Please be sure your work is readable (either written clearly or typed).

Problem 1. Binary Search Trees

This exercise is about ordinary binary search trees, not balanced search trees. That is, use the algorithms of Figures 4.22 and 4.25 (these figures are the same in both the 2nd and 3rd editions of the textbooks).

(a) Show the binary search tree that results from inserting the keys 186, 039, 991, 336, 778, 066, 564, 154, 538, 645 and 256, in this order, into an initially empty binary search tree. You need only show the final tree.

(b) Show the result of deleting the key 186 from the final tree of part 1a. In the event of deleting a node with 2 children, replace it with its successor (the minimum value child in the right tree), and not its predecessor.

Problem 2. AVL Trees

(a) Find the AVL tree that results from inserting the keys 186, 039, 991, 336, 778, 066, 564, 154, 538, 645 and 256, in this order, into an initially empty AVL tree. Show the AVL tree after each insertion (including possible rotations) is completed.

(b) Show the result of deleting the key 186, 336 and 538 from the final tree of part 2a. In the event of deleting a node with 2 children, replace it with its successor (the minimum value child in the right tree), and not its predecessor.

Problem 3. AVL Verification

Give pseudocode for a linear-time algorithm that verifies that an AVL tree is correctly maintained. Assume every node has fields key, height, left, and right and that keys can be compared with <, ==, and >. The algorithm should verify all of the following:

(a) The tree is a binary search tree.
(b) The height information of every node is correct.
(c) Every node is balanced.

Your pseudocode should throw an exception if one of these properties is violated; if the tree is a valid AVL tree, no exception should be thrown.