CSE 373 Data Structures Spring 2016

Practice Problems

Problem 1

In this problem you will practice insertion into binary search trees and AVL trees.

A Show how to insert 4, 6, 9, 2, 3 and 7 into an initially empty binary search tree. (Show each step.)

B Show how to delete the root from the binary search tree you created. Use the successor node to replace the root. (Show all work.)

C Show how to insert 4, 6, 9, 2, 3 and 7 into an initially empty AVL tree. (Show each step, including rebalancing.)

Problem 2

In this problem you will practice insertion and deletion in binary heaps (default min heap).

A Show how to insert 10, 12, 14, 6 and 1 into an initially empty binary heap. Insert each value, one at a time (not with buildHeap), and show each of the 5 steps as separate trees (pictorially with nodes and edges). For only the step of adding the 1, show the initial array representation and each step of the percolate up until the 1 is in the right place.

B Show how to build a binary heap with 10, 12, 14, 6 and 1 using buildHeap algorithm. Show each step as a separate tree (pictorially with nodes and edges).

C Show the results of a deleteMin operation on the heap above. Show the initial array representation and each step of the percolate down until the operation is complete.
Problem 3

In this problem you will practice working with the union-find algorithms and up-tree data structure. You are given 6 individual sets numbered 1 through 6. Show the results of the following sequence of instructions (show each step as a tree):

union (1, 2), union (3, 6), union (4, 3), union (4, 5), union (1, 4)

when unions are:

A Performed arbitrarily by making the second argument a child of the first argument.

B Performed by size.

Problem 4

Show how to insert the following keys into a B+-tree. The tree is a 2-3 tree, meaning that each internal node has 1-2 keys and up to 3 children. Each leaf node should also have at most 2 keys for this problem.

Keys to insert in this order:
82 96 53 46 91

Show all 5 steps in construction of the B+-tree. And note that when a leaf has to split, because it would contain 3 keys, move ONE key to the left leaf and TWO to the right to obey the algorithm I gave in class. If you do it the other way, it will be wrong, even though it is reasonable.