COSC229 Assignment 1  Data Structures and Sorting

(1) Generate 10 random numbers of your choice that satisfy the requirement of (2). Insert them into a binary search tree (BST). Then delete the root twice. For the trace, draw the tree after each insertion. Show the path from the root to the insertion point. Pictures can be hand drawn if neatly done. For deletion, show the tree after restructuring. Confirm the tree structure after ten insertions by the program for (3) with tree enhancement. [5 points]

(2) Insert the same numbers into an AVL tree. There need to be at least three rotations. Give the tree after each insertion, and after rotation if rotation is required. Pictures can be hand drawn if neatly done. Balance labels (L, E, R) must be shown. Confirm the tree structure by the program for (4) with tree enhancement. Then perform a delete operation that requires rotation. [15]

(3) Do computer experiment of inserting n random numbers into a binary search tree where n should take ten different values from 10000 to 1000000. Measure the number of key comparisons and cpu time. For cpu time, counting key comparisons should be removed. For large experiments, the modulus for the random number generator must be large. Remove tree enhancement for this experiment. [10]

(4) Do a similar experiment with an AVL tree, and compare performance with (3). For the AVL program, complete an unfinished program attached. Remove tree enhancement for this experiment. Also give comments on each line, following an example for BST attached. [20]

(5) Do the same experiments with (3) and (4) with an increasing sequence of input numbers. [10]

(6) Now enhance the programs for BST and AVL tree with the in-order traversal routine, so that they can be used for sorting. Do experiments with the following sorting algorithms with random numbers: Enhanced BST, Enhanced AVL, minimum selection sort, quicksort. For experiments, measure cpu time and key comparisons. n should be the same as before, but stop earlier for minimum selection sort. [20]

(7) Discuss experimental results by comparing performance of those methods. Also discuss possible improvements on the AVL code. [20]

Note. Tree enhancement is in-order traversal plus “(“ and “)“.

Submit the following
Hard copy for (1)--(7) above.
Electronic submission: Source programs. bst.c, avl.c, quick.c, min.c
Bst.c and avl.c with tree enhancement, and quick.c and min.c with output statements.
Due date: August 19, 5pm, drop due August 26, 5pm