WASHINGTON STATE UNIVERSITY

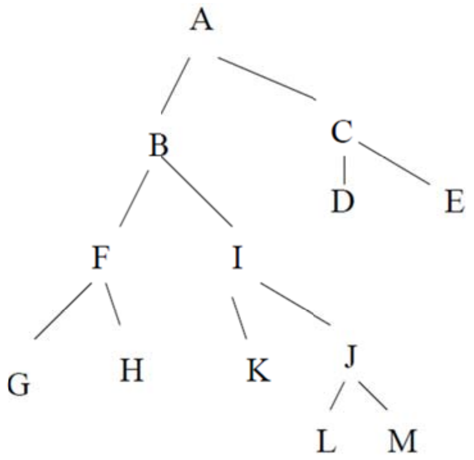
School of Electrical and Computer Engineering
CptS223: Advanced Data Structures C/C++
Spring 2018
Homework #3
Due: Monday 02/12/2018 @ 11:59pm

Instructions: This is an individual assignment and all work must be your own. Be sure to show your work when appropriate. This assignment must be turned in by the due date via Blackboard in *PDF* format. You may use any editor you like (or even print it out, *legibly* write in answers, and scan it in), but convert it to *PDF* for submission.

A maximum of 24 hours delay is permitted at the cost of 10% penalty. Beyond this 24-hour delay, not submissions will be graded. That is, assignments submitted after 24 hours after the above due date will not get any credits.

1. Give the expression tree for ((a+(b-c))*(d-e))+(f*(h-i)).

2. For the tree shown in this figure, answer the following questions.
   a) Height of the tree = ?
   b) Depth of node J = ?
   c) Height of node J = ?
   d) Redraw the tree by the First-child, Next-Sibling method.
   e) Give the post-order, pre-order and in-order traversals of the tree.



3. It is possible to reconstruct (meaning, draw) the tree given only the tree's pre-order and in-order traversals. To illustrate this, draw the tree by reconstructing it from the following traversals (note that both traversals represent the same tree):
   Pre-order traversal: a, f, e, d, g, b, c
   In-order traversal: e, f, g, d, b, a, c

4. This question has two parts:

   a) Draw the BSTs that result from the following two different insertion sequences of the same set of elements:

      Insertion sequence: 8 7 1 6 2 3 5 4

      Insertion sequence: 5 7 3 2 8 6 4 1

      For both cases, start with an empty tree. Just showing the final tree in your answer is sufficient. Alternatively, if you want to show the BST after every insertion, that is fine too.

   b) Briefly state what is so markedly different between the shapes of the two resulting BSTs.

5. Starting with an empty tree T_0, show the set of BSTs $T_0 \Rightarrow T_1 \Rightarrow T_2 \Rightarrow \dots \Rightarrow T_6$ resulting from performing the following sequence of operations (in that order): Insert(5), Insert(10), Insert(2), Insert(7), Insert(6), Remove(5).

6. This question has two parts.

   a) Starting with an empty tree T_0, show the set of AVL trees $T_0 \Rightarrow T_1 \Rightarrow T_2 \Rightarrow \dots$ resulting from performing the following sequence of operations (in that order): Insert(7), Insert(5), Insert(2), Insert(4), Insert(3), Insert(1). If at any step you need to rebalance the tree using rotation, then clearly identify: (i) the node that has the imbalance (i.e., violation), and (ii) the corresponding rotation “case” that applies there (i.e., case 1 or 2 or 3 or 4).

   b) Answer the same question as in part (a) but with this insertion sequence instead: Insert(2), Insert(1), Insert(4), Insert(5), Insert(9), Insert(3), Insert(6), Insert(7).