CSCI 220 – PA 9

Binary Trees

Feel free to discuss and help each other out but does not imply that you can give away your code or your answers! Make sure to read all instructions before attempting this lab.

You can work with a lab partner and each one must submit the same PDF file (include both names in the submission file). Each person must include a brief statement about your contribution to this assignment.

You must use an appropriate provided template from Canvas or my website (zeus.mtsac.edu/~tvo) and output "Author: Your Name(s)" for all your programs. If you are modifying an existing program, use "Modified by: Your Name(s)".

Exercise 1: Set up **LinkedBinaryTree** class (must use C++ code fragment or Java code, but make sure to understand the code) and then create a simple test driver to perform some basic operations on a binary tree of strings. Once it is working correctly, modify LinkedBinaryTree to support post-order traversal. It is best to modify positions() to perform post-order traversal.

Test Case 1: Add code to construct a binary tree below and then perform a post-order traversal on that binary tree. Confirm it is working correctly.

```
            A
          /   \
        B       C
      /   \
    D       E
```

Post-order traversal for binary tree above: D E B C A

Test Case 2: Remove node E from the binary tree above and then perform a post-order traversal on that binary tree. Confirm it is working correctly.

Post-order traversal for updated binary tree without node E: D B C A

Note for C++: LinkedBinaryTree class is implemented as a proper binary tree and only internal nodes hold data. Current expandExternal does not accept a value so it is best to *overload the function expandExternal by adding a second parameter for value to be added: void expandExternal(const Position& p, Elem &e);*

Note for Java: ***LinkedBinaryTree*** class *is implemented as an improper binary tree and all nodes hold data.*

Question 1: Provide pre-order traversal, in-order traversal, post-order traversal, and level-order traversal for the following binary tree.

```
            A
          /    \
        B        C
      /   \        \
    D      E         F
          /
        G
```

Question 2: Which two traversals are useful for an arithmetic expression tree? Explain.

Extra Credit: Modify LinkedBinaryTree to support different types of traversals such as preorder, in-order, post-order, and level-order traversal. It is best to overload positions() so it can perform appropriate traversal (use int as a parameter). Add two more traversals for this part (pick two of the following: preorder, post-order, or level-order traversal) so your binary tree supports three different traversals.

Fill out and turn in the PA submission file for this assignment (save as PDF format).