COP 3502 Suggested Program Edits: Binary Search Tree Insert, Tries (Week 8 Programs)

1) In the posted code bintree2.c, there is a function numNodesOneChild, that returns the number of nodes in a binary tree with one child. Write a similar function, numNodesTwoChildren, that returns the number of nodes in a binary tree with two children.

2) The goal of bintreeexample.c is to find the average height of a tree created with a million random inserts. Another item of interest would be the sum of the differences between the left and right subtrees of every node. Add a function to bintreeexample.c that takes in a pointer to the root of a tree and returns the sum of these differences. If a node has a null child, we consider the child's height to be -1.

3) One way to deal with tied elements in a binary tree of integers is to store two pieces of information in the node - the value that is being stored and its frequency, the number of times it has occurred. This way, in the tree, all items to the left of a node will be strictly less than it, and all items greater than a node will be greater than it. Edit bintree2.c to work this way.

4) Add a function to any of the trie examples that counts the number of different nodes in a trie and returns this value.

5) Often times, there are common suffixes of words, such as "ing." One might want to count how many words have a particular suffix. Find a way to edit the example countprefixes.c so that if counts the number of words with a particular suffix. (Hint: You won't actually change any of the code in the numPrefix function at all!!! Instead, you'll store something a bit different in the trie and search for something a bit different in it.)