

CS 140A - Assignment 3

Due Thursday, November 13

In this assignment you will be implementing a binary search tree. You must use a linked-list to implement the tree. Each node in your tree should hold a character. Each node will also contain an integer called *count*. Include the following functions for the tree.

Insert Insert an item into the binary search tree. Be sure to keep the binary search tree properties. When an item is first inserted into the tree the *count* should be set to 1. When adding a duplicate character, rather than adding another node, the *count* variable should just be incremented. You should insert in a case-insensitive manner. In other words, the letter 'a' and the letter 'A' would be considered as the same letter.

Search Write a recursive function that searches for a character in the binary search tree. It should return a pointer to the node in which it is found or it should return NULL if the character is not in the tree. Again, this should be case-insensitive.

FindMin Find and return the smallest character (the one with the smallest ASCII value) in the tree.

FindMax Find and return the largest character (the one with the largest ASCII value) in the tree.

Height Compute and return the height of a particular subtree. This function should take a pointer to a node and return the height of the subtree using the specified node as the root. Thus, if the root was passed to this function, the value returned would be the height of the entire tree.

Remove Remove a node from the tree. Be sure to maintain all binary search tree properties. If removing a node with a *count* greater than 1, just decrement the *count*.

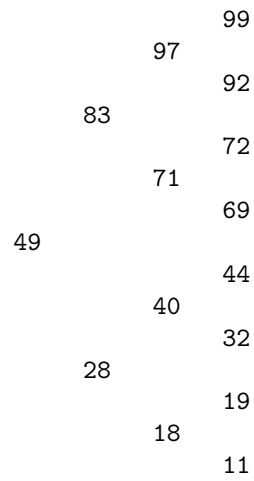
Print Print the tree in the following manners. When printing a node, print the character followed by the count in parentheses. For example, $c(2)$, $k(1)$.

Preorder Traverse and print the tree in preorder notation. Write this function recursively.

Inorder Traverse and print the tree in inorder notation. Write this function recursively.

Postorder Traverse and print the tree in postorder notation. Write this function recursively.

Tree Illustration Print the tree such that it illustrates the structure of the tree in a sideways fashion. For example, the root of the following tree is 49 with the left child being 28 and the right child being 83.



Submit your program electronically using the instructions provided by the TA. Be sure to use good programming style and include meaningful, thorough comments. Remember that no credit will be given if your program does not compile.