John Gill

Dr. Tala Talaei Khoei

CS 5008

25 July 2024

Lab Activity 4

This lab activity focused on binary trees – making them, and analyzing them. The first question calls on us to make a binary tree, then inspect it for total nodes and leaf nodes. The total nodes are counted recursively by adding one to the count, with left and right subtrees included. Then, for counting leaf nodes, the function iterates through each node in a given tree, checks if it has children, and if it doesn’t, adds one to the counter until all nodes are checked. Question 2 has us make a program that determines if a tree is a BST, which was done by checking if each node’s data is within the proper range, meaning the left subtree of a node contains only nodes with keys less than the node’s key, and the same for the right subtree, except all keys in the right subtree are greater. Question 4 had us inspect two trees to see if they were mirrors of one another. I noticed an interesting pattern here that child nodes just need to be initialized in opposite order for it to work. I think all child nodes can be children of the same parent nodes between two mirrored trees, and that the rule of initializing them in flipped order is universal. Here, we use recursion to make sure left is equal to right, and right is equal to left between two binary trees. And for question 5, we’re tasked with the duty of determining if two trees are identical. That was done simply by recursively iterating through two separate trees simultaneously and making sure that each node is identical at each time. If any two nodes between the two trees were not identical, it immediately ends the recursive loop and says they are not.