Binary Search Tree Algorithm

Homework #7

By

John Kiyak

CS303

October 16, 2018

**Problem Specification**

The goal of this assignment was to write a program that would implement a binary tree algorithm recursively. You were supposed to make methods for inserting nodes into a tree, printing out the values as the tree gets “walked”, and also a search method to search for a specific key value in the tree.

**Program Design**

For this program, the main function reads and stores data from ‘input.dat’ and ‘UPC.csv’ into separate arrays. Then, a key is chosen using the random.choice function, from the ‘input.dat’ file, and store into the key variable. Then, the key is printed. The insert function insert the chosen variable into the node. The inorder class prints out the tree if it is in order. The search function takes in the root of the tree, and the key that is being looked for, and returns either the root, or continues searching until the key is found in the root.

**Testing Plan**

For my testing plan, I would make sure that the arrays for both the ‘input.dat’ and ‘UPC.csv’ files were printed and stored correctly. I would then make sure that the key is printed and different for every test. Afterwards, I would make my node, and insert different values using the insert command, make sure that the inorder function works, and verify that the search function works as intended.

**Analysis and Conclusions**

Considering that a binary search tree has a normal time complexity of O(h), where height is the height of the binary search tree, and at worst case it has a time complexity of O(n). Considering the length of this search tree would be very high from the input file (over 150000 entries in the UPC.csv file), much time would be needed to search and sort a binary search tree of that size.