# California State University, Fresno

# DEPARTMENT OF COMPUTER SCIENCE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Class: | **Algorithms & Data Structures** | | | Semester: | **Fall 2021** |
|  | | | | | |
| Points |  | Document author: | **Saishnu Ramesh Kumar** | | |
|  | Author’s email: | **saishnu\_rk@mail.fresnostate.edu** | | |
| Laboratory number: | **Section 1, 11am to 12:50pm** | | |
|  | | | | | |

**1. Statement of Objectives**

This lab assignment dealt with creating a Binary Search Tree or BST for short. We were tasked to create a BST that would insert a node, find a node, and to find the largest node. The importance of this assignment is to understand how do BSTs work and how are you supposed to traverse through them to get to a specific answer. There are different methods to traverse the tree, the first being the post order where it starts from left, right, and the root. The second one being pre-order where it begins from the root, left, and then right. The third one would be the in order method where it starts from the left, root, and finally the right. The tree traversal is O(n) where n stands for the size of the tree. The running time of basic operation on the BSTs are Big Theta(log n). This report covers what was done for the code in regard to the BSTs.

**2. Experimental Procedure**

For this assignment, I followed the header file that was given and created a number of functions that are within the BST class. A constructor was created, followed by a getRoot Function that would return the root. The printTree function will recursively called both sides of the tree, the left and right child. The insertNode will also recursively call itself to help insert a new node into the tree. The findNode function will help to search for a specific element in the list/tree and if it’s unsuccessful, it will return NULL. The Largest function will be returning the the largest element that is found in the tree. There is also a print function that will print out the elements in the list as well as a insertion function, findLargest function, and searchNode function that helps with outputting the specific parts of each function by calling them into these ones. In the main function the outputs: Post-order, Largest element, and Search Node.

**3. Analysis**

Screenshot Terminal Output:

Text

Description automatically generated

It can be seen that the post order output is shown, with the largest element in the list found as well as the search functions looking for the elements that have checked to see if they exist in the BST.

**4. Encountered Problems**

I normally came across a number of compiler errors and a dumping stack error once but I was able to fix the issue.

**5. Conclusions**

From this lab, I have learnt a little more about BSTs and how they operate and move along its tree to get to a specific number as well as knowing how to search for a specific node, inserting a node from the bottom of the tree, finding the largest node in the list, and to print the list in a post-order traversal format.

**6. References**

Slides provided by TA

<https://www.geeksforgeeks.org/binary-search-tree-set-1-search-and-insertion/>

<https://cppsecrets.com/users/19310910011510410111111497110485764103109971051084699111109/C00-program-to-insert-an-element-into-binary-tree.php>

<https://www.geeksforgeeks.org/binary-search-tree-set-1-search-and-insertion/>