Implementing Binary Search Tree
Lab #7

By
Joshua Matthews

CS 303 Algorithms and Data Structures
October 14, 2015

1. Problem Specification

The goal of this Lab assignment was to implement a Binary Search Tree and to create functions to insert, search, print, and delete nodes inside the structure itself.

2. Program Design

This program required two generic classes. One called BinaryTreeNode to hold information and methods to manipulate the node. The second class was called BinaryTree and it stored one BinaryTreeNode and would manipulate and search for all other nodes through that one node.

The following steps were needed to develop this program:

A. I setup the two classes: BinaryTreeNode and BinaryTree.
B. I created them both template classes so they could be generic and take on any type they needed to be.
C. I gave the BinaryTreeNode a generic variable property called value to hold it's value. Then I gave it two other BinaryTreeNode properties to hold the left and right child nodes to itself.
D. I setup the methods to set and get the two child nodes.
E. Inside the BinaryTree class I gave it a BinaryTreeNode property to be used as the head of the tree.
F. I implemented the insert, search, print, and delete methods for the class and created any helper functions to go with those to make tasks a little bit easier to separate.
G. The insert method compared the value to every other nodes value until it reached a NULL node and would create a BinaryTreeNode object set with it's value and place it in the empty node. If it came across itself it would do nothing.
H. Search was similar in the way it found the object and then just returned it.
I. The print method went all the way down the left side of the tree first and then started printing every node until it reached to far right node.
J. The delete method was the most difficult to implement. I had to setup test cases for if both child nodes were empty or if just one of them were empty and if neither were empty we had to do some searching and rearranging of nodes.

The included libraries were: iostream(for printing out to the screen).

3. Testing Plan

The Lab07.cpp file tests our implementation of a Binary Search Tree. I setup the tree in the main method by instantiating it and then inserting 8 values in of random different numbers and outputted to the screen each time I inserted a new value. I called the search method and saved the return value to a local variable and then outputted to the screen the local variables value to show that the returned object was the correct object. I then call the delete method to delete the element that would be the head in the BinaryTree to show that the nodes were correctly rearranged and I also deleted the far left node to show that it would work for that as well. I then printed out the tree in array form to show all elements that were in the tree.

4. Test Cases

Setup Code	Output

auto tree = new BinaryTree<int>();
cout<<"Insert "<<5<<endl;
tree->insertNode(5);
cout<<"Insert "<<3<<endl;
tree->insertNode(3);
cout<<"Insert "<<7<<endl;
tree->insertNode(7);
cout<<"Insert "<<2<<endl;
tree->insertNode(2);
cout<<"Insert "<<9<<endl;
tree->insertNode(9);
cout<<"Insert "<<1<<endl;
tree->insertNode(1);
cout<<"Insert "<<4<<endl;
tree->insertNode(4);
cout<<"Insert "<<0<<endl;
tree->insertNode(0);
auto five_node = tree->searchNode(5);
cout<<"Searched for 5, function returned = "<<five_node->getValue()<<endl;
auto zero_node = tree->searchNode(0);
cout<<"Searched for 0, function returned = "<<zero_node->getValue()<<endl;
 	
cout<<"Deleting Node 5"<<endl;
tree->deleteNode(5);
cout<<"Deleting Node 0"<<endl;
tree->deleteNode(0);
cout<<"Printing tree: ";
tree->printTree();
cout<<endl;
delete tree;
	Insert 5
	Insert 3
	Insert 7
	Insert 2
	Insert 9
	Insert 1
	Insert 4
	Insert 0
	Searched for 5, function returned = 5
	Searched for 0, function returned = 0
	Deleting Node 5
	Deleting Node 0
	Printing tree: |1|2|3|4|7|9

5. Analysis and Conclusion

The Binary Search Tree is a wonderful data structure that will keep all of your elements in order and have fairly good insertion and deletion times as long as the head of the tree is set to a value in your data that would be the middle value. If the head is the first value of your data then it will be like using an array of objects.

6. References

The www.cplusplus.com has references and documentation on all of the standard libraries that were used in this program.