Willem Kroeger

CS 300: Data Structures and Algorithms: Analysis and Design

Southern New Hampshire University

February 5, 2023

Writing the code for the binary search tree came easier than the linked list and hash tables we worked on previously. Having to simply decide between two options, left or right, to traverse the tree was simple to understand and implement. Not all methods are used in the final product, as the menu doesn’t have options to use them.

The binary search tree looks for bids based on the bid ID number. If it is smaller than the “current” number, it shifts the number to the left and vice versa for the right. Public and Private methods to call for preorder and post order sorting and printing needed to be added to ensure they would work with the current structure, and the removeNode method needed to be created.

**Insert bid**

If root is empty  
 create new node

Else if root is not empty  
 **add node**

**Search bid**

While current node isn’t null  
 if current node matches bid id, return bid  
 if current bid is smaller, traverse left tree  
 else, traverse right tree

**Add Node**

if bid to be added is smaller than current node  
 Traverse left tree  
 if node-> left is null, add node

if bid to be added is larger than current node  
 traverse right tree  
 if node->right is null, add node

**Remove Node**While current node is not null

If current bid is less than bid to be removed  
 Traverse left tree  
 if current bid is greater than bid to be removed  
 Traverse right tree

If current bid equals bid to be removed

if left and right pointers are blank ( both leaves empty)  
 delete current bid

if right pointer is blank  
 set temp bid to current bid  
 set current bid to point to right pointer  
 Delete temp bid

if left pointer is blank  
 set temp bid to current bid  
 set current bid to point to left pointer  
 delete temp bid

if both pointers are full (both leaves have children)  
 set temp bid to current bid’s right pointer  
 while temp bid left pointer is not null  
 search left tree  
 set current bid pointer to temp bid pointer  
 recursively send current bid right pointer into removeNode (recursively remove current node and adjust tree structure)