**5-2 Assignment: Binary Search Tree**

Aaron Ciminelli

Information Technology Adjunct Faculty, SNHU

CS-300-T6609 DSA: Analysis and Design 22EW6

**Wasim Alim, MSEE**

**July 31, 2022**

* **Code Reflection:** While developing this code, I encountered several roadblocks. It seems as though there were missing sections outside of the FIXMEs. I had to look back on previous assignments and use the tutors for additional help. For instance, I had to add the preOrder and postOrder data members and methods. This project aimed to search through a data set and make changes systematically.

|  |  |
| --- | --- |
| CHOICE 1: | Text  Description automatically generated |
| CHOICE 2: | Text  Description automatically generated |
| CHOICE 3: | Text  Description automatically generated with low confidence |
| Choice 4: | Text  Description automatically generated with low confidence |

* **Pseudocode or Flowchart:**

**BinarySearchTree::BinarySearchTree()**

**Set root as null**

**Void BinarySearchTree InOrder()**

**Call the inOrder function and pass the root**

**Void BinarySearchTree PostOrder()**

**Call function postOrder(root) and pass root**

**Void BinarySearchTree PreOrder()**

**Call function preOrder(root) and pass root**

**Void BinarySearchTree Insert(Bid bid)**

**If the root is equal to null**

**The root is equal to the new node bid**

**Else**

**Call function addNode(root, bid)**

**Void BinarySearchTree Remove(str bidId)**

**call remove node root bidID**

**Bid BinarySearchTree::Search(string bidId)**

**Set current Node equal to the root**

**While (currNode is not null)**

**if a match is found, return the current bid**

**Return current bid**

**If bid is smaller**

**Traverse left**

**Else**

**Traverse right**

**Return**

**void BinarySearchTree::addNode(Node\* node, Bid bid)**

**if Node is larger, then add to left**

**If no left Node**

**Node becomes left**

**Else**

**If no right node**

**Node becomes right**

**Else**

**Recurse down the left Node**

**void BinarySearchTree::inOrder(Node\* node)**

If Node is not equal to null

inOder left

output bidID, title, amount, fund

inOrder right

**void BinarySearchTree::postOrder(Node\* node)**

If Node is not equal to null

postOrder left

postOrder Right

output bidID, title, amount, fund

**void BinarySearchTree::preOrder(Node\* node)**

**If Node is not equal to null**

output bidID, title, amount, fund

**preOrder left**

**preOrder right**

**References:**

Vahid , F., Lysecky, S., Wheatland, N., Siu, R., Lysecky, R., Edgcomb, A., & Yuen, J. (2019). *CS 300: Data Structures and Algorithms*. Zybooks. Retrieved July 31, 2022, from https://learn.zybooks.com/zybook/CS-300-T6609-OL-TRAD-UG.22EW6/chapter/6/section/1