Augustus Mendy

CS-300-11373-M01

Southern New Hampshire University

June 2, 2024

5-2 Assignment: Binary Search Tree

Reflection

This week's task was to finish writing the code that would enable the binary search tree to read, add, and remove items from a list. Values larger than the root are positioned to the right in binary search trees, whereas values less than the root are positioned to the left. With higher values on the right and lower values on the left, the process will repeat itself along each leaf. This will cut down on the amount of time it takes to search the list because you can locate the value within half of it (to the left or right) after comparing it to the root value. The assignments seem to be becoming easier, in my opinion. This one was quite easy to follow and quite basic. Generally speaking, whenever I run into difficulties, I consult ZyBooks, which typically helps me get through them.

Pseudocode

Create class BinarySearchTree

Define methods for Insert, Print, Remove, Search

Create a root that points to null

Create Insert method

If Root is Null, new item is Root

Else is item is less than root, add left

If left is null, add item

Else

if item is less than leaf add left

if item is greater than leaf add right

Else if item is greater than root, add right

If right is null, add item

Else

if item is less than leaf add left

if item is greater than leaf add right

Create Print Method

If root is not null

Traverse left, output if found

Traverse Right, output if found

Create Erase Method

If root is empty, return

Else, traverse left, if node found delete, set to null pointer

Else, traverse right, if node found delete, set to null pointer

Create Search Method

If root is not null

Traverse left, return if found

Traverse Right, return if found