Journal Assignment

* Binary tree algorithms can be complicated: Are they worth the effort? Recall the characteristics of sort algorithms that use a binary “divide and conquer” technique.

Module 6 we went in depth with how Binary Search tree algorithms are structured and implemented for different functions. I can agree that it does become very busy and the utilizations of recursive and while loops along with starting from the root and passing into nodes with the specified functions makes code look a bit intimidating and can confuse the person developing it. Based on running the code in the command prompt I can assess that the BST can beat an array in the number of operations it takes to insert or delete an element. Unlike an array or the sort algorithms you need to shift elements for the functions implemented whereas BSTs don’t require its elements to be stored in contiguous block of memory. You can adjust pointers for inserting or deleting a node. The complication this week in dealing with BST is the fact that each node is a struct containing data and pointers to a left child and right child not to mention the internal manipulation took me time to understand. Ultimately, I can say that yes they are worth the effort for searching exact data unlike sort algorithms an “element access can be done in constant time... If we were to use a BST in the same scenario there would be no way to get to the element without starting at the root node and traversing to the designated element.” (Biguenet, 2016).

* If each node of a tree is large (i.e., contains a large amount of data), then available memory may be a limiting factor. What can be done to allow an extremely large tree with hundreds of thousands (or even millions) of nodes to be built and managed?

For BST that have a hundreds of thousands of nodes to be built and managed we would have to implement those class members just like in this week’s exercise to be able to have insertions and removals for any internal and child notes that extend form the root. This is a way to be able to build and manage a BST of that size.

**Resources:**

- Biguenet, J. (2021, May 13). How and when to implement binary search trees. LookFar.

Retrieved February 12, 2023, from <https://www.lookfar.com/blog/2016/07/28/why-binary-search-trees-matter/>

ZyBooks Chapter 6