Binary Search Tree

The task was to implement Binary Search Tree with array as a base structure. The required operations were the following:

• Insert value into the tree

• Search value into the tree

• Find minimum value in tree

• Find maximum value in tree

• In order traverse

• A traverse for the order before

• A traverse for the order after

I decided to use concept of vacant nodes in tree. During all searches and orders vacant nodes help ignore, so the tree looks like it really does not contain deleted values. All operations done had to be minorly changed, for example, if during insertion operation we found vacant node with inserting value, we can just mark node as it is no more vacant. For testing of implementation, I created test driver file mainBST.cpp, which creates tree with values from 1 to 10, inserted in shuffled order.

Max Heap

Max heap is implemented in classic manner: besides standard operations of push (this will insert a value into heap), pop (this will remove max element from heap), peek (this will get max element from heap), clear (removes all elements from heap), I have implemented method heapify, which accepts array of integers, and creates MaxHeap out of these values. For this part of my code, I worked with a classmate, Cayman Hubbard, on this part of the assignment. For testing of implementation, I created test driver file mainHeap.cpp. To validate implementation, I have also completed bonus part, which requires implementation HeapSort. The driver file sorts the values of shuffled array. The correctness of sorting provides validity of MaxHeap implementation.

Compile & Running

To compile my program, I used terminal and ran the command “g++ mainBST.cpp binaryTree.cpp binaryTree.h. -o <execution name>” (same thing but with Heap file name for Heap testing.) I did not realize that we were to work with a csv till it was late, but I believe you could test it in terminal with piping. Both should be tested and compiled separately. I also created a Test file where one could input numbers then after imputing all numbers putting a letter then typing run after imputing the numbers, this would help run the code with the users own discretion of numbers.