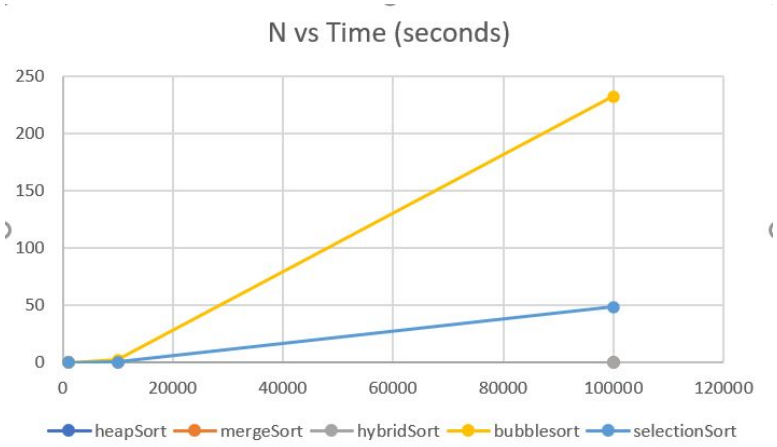
Question 1: make q1

a) For isSorted I assumed that the vector was sorted if it was in increasing order. The complexity of this algorithm is O(n) since we only loop through vector once.
b) For bubbleSort I assumed again that it was sorted if it was increasing order. I made a recursive method that will pass through the vector and swap items until the vector is in increasing order. The time complexity is O(n^2) since we have a for loop in a recursive method;
c) For selectionSort we have 2 loops, 1 loops through the entire vector while another loop in that loop finds the index of the smallest element after the current element. Then if the minimum found is smaller than the current element, then they are swapped. This also has a time complexity of O(n^2) since we have 2 nested loops.
d) For hybridSort, we divide the list into 4 sections, since the size might not be divisible by 4, we let the last sublist store the remaining elements. Therefore there will be 3 elements of the same size with the last sublist containing the extra elements. I was not able to merge the sublists, but I did call quicksort on the entire vector after it was sorted with the 4 sublists did is probably very inefficient. If the list is less than 16, then bubbleSort is called.
e)



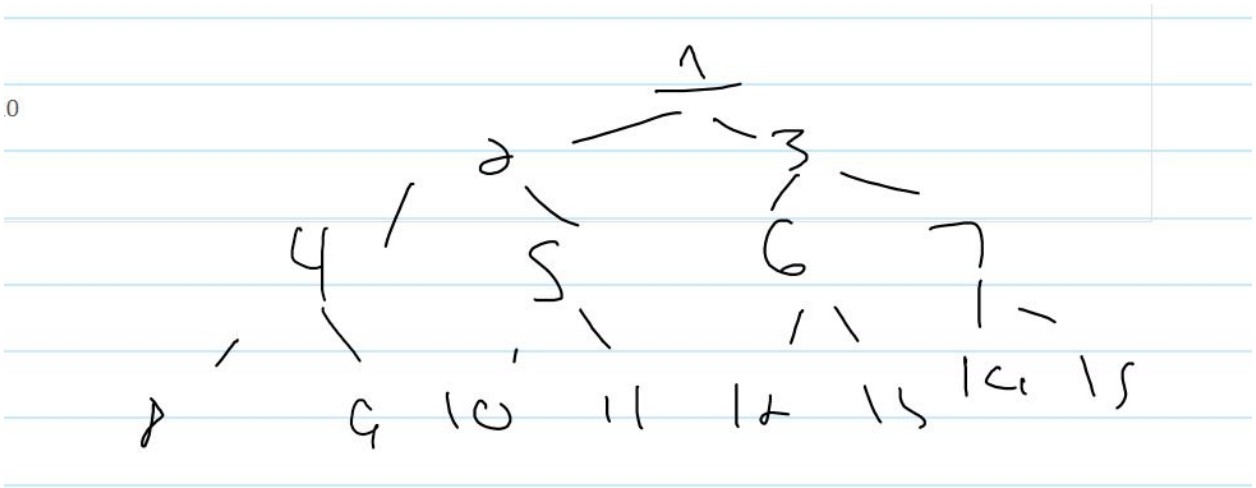
As seen in the graph, all but selectionSort and bubbleSort had a time complexity of under 1 second for n values under 100,000. As shown in the graph an O(N^2) complexity results in a massive increase in time.

Question 2

For sections A use command Make q2a

a) I was not able to reconstruct the tree after changing, or deleting elements. I was able to get

b)



This has a time complexity of nlog(n)

c) Use command make q2c for section C

The kthsmallest method creates a mini heap from the given input array. Then it will extract the min k times to find the kth smallest. My mini heap from part A does not work, so this program will display incorrect values. This should have a time complexity of O(nlon(n)) since it uses a loop to create the mini heap, and inserting in the mini heap takes log(n) times. Therefore the method has a time complexity of O(nlog(n))