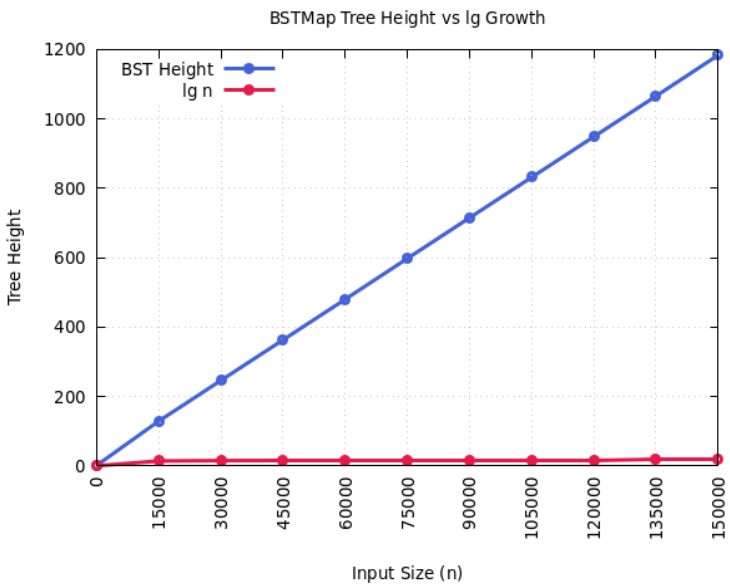
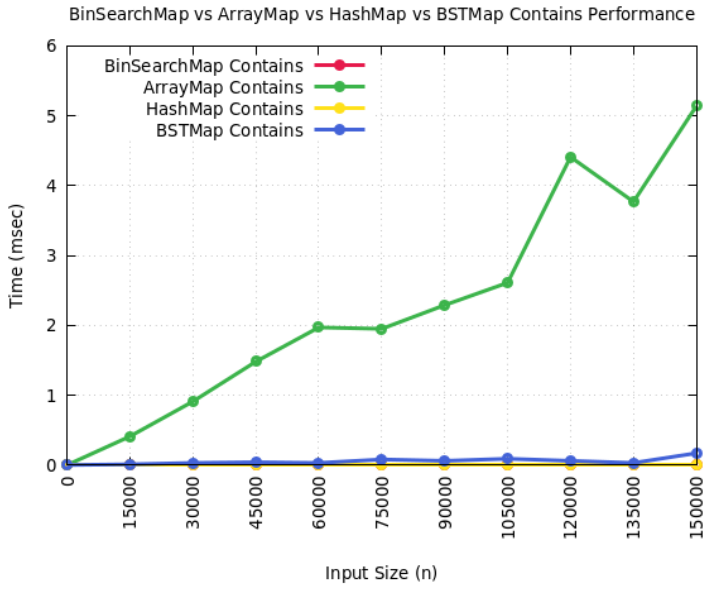
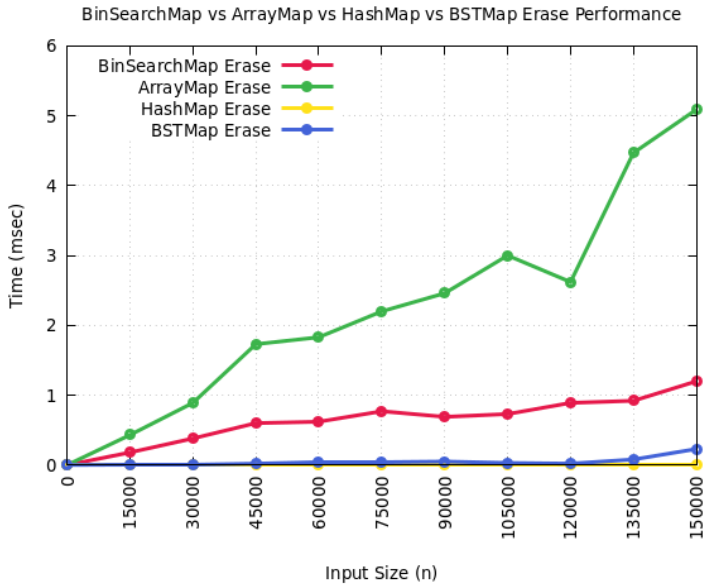
Dominic MacIsaac
CPSC 223
Dec 2 2021
HW7 Results



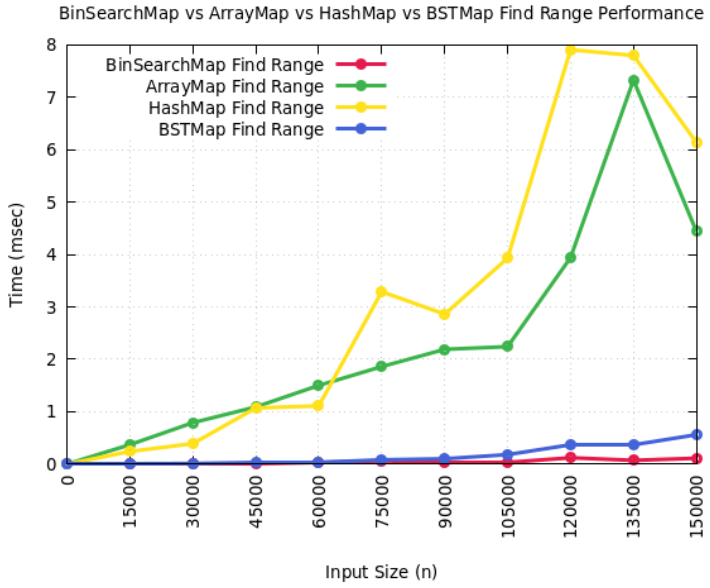
This BSTMap is unbalanced and therefore has a Big-O worst case of O(n). This is because if every new value is bigger than the root(or smaller), the tree acts like a linked list. This, as shown by this graph, is much less efficient than log n.



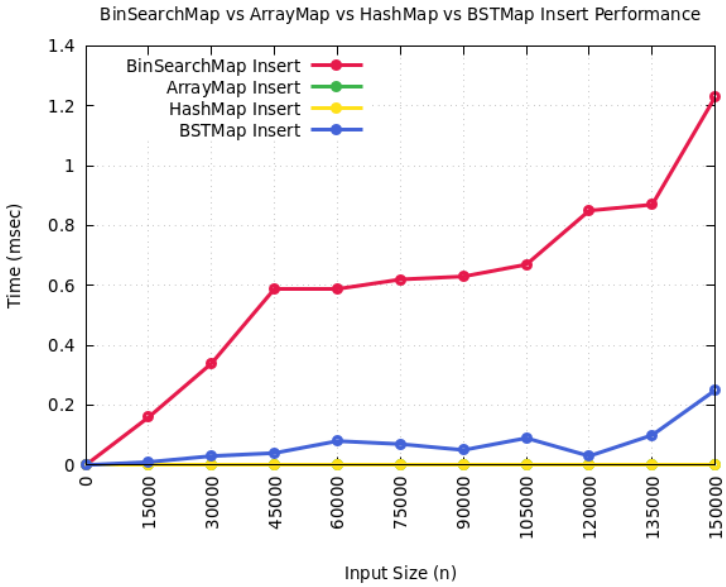
Array Map is the only of the four that doesn't use any advanced algorithm to check contains. This is why the search is O(n) compared to the others. The rest are roughly 1 to log n because their algorithms allow them to find where the value of contains would be.



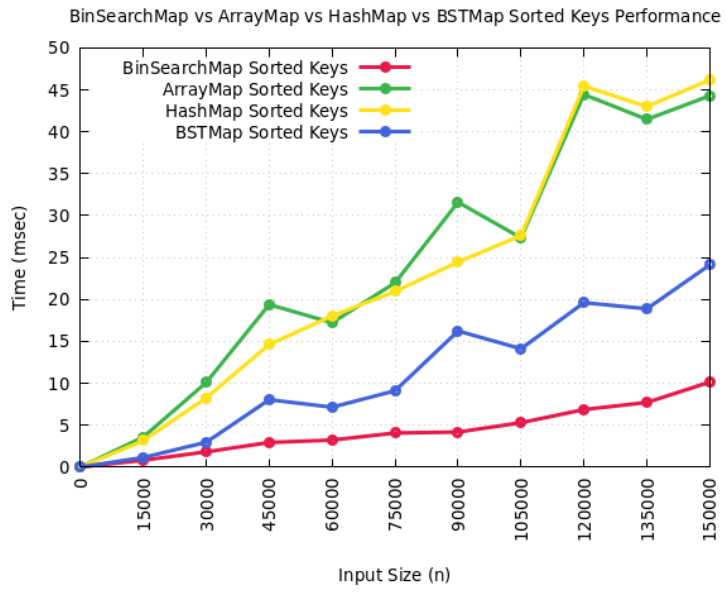
Hashmap and BSTMap are both extremely quick because they can locate the value to erase extremely quickly. BSTMap is slightly slower though because it has to copy and change values if a parent with 2 children is deleted. Arraymap is slowest because of its difficulty finding the value (as seen in contains)



Hashmap and Arraymap have to iterate through every value which is why both of their graphs are very slow compared to the other two. Binsearch is quick because it can just find the k1 and then iterate through to k2 since they are in order. BSTmap utilizes the fact that it is ordered by bigger and smaller to only recursively visit the nodes it needs to visit and it extracts everything smaller than k1 and bigger than k2.



Bin search takes so long to insert because it has to place the item in the correct position and then shift the values following. It also has to resize every time the array gets filled. Hashmap and array map are so quick because they can place the value nearly immediately where it belongs (though hashmap sometimes must iterate through a few linked list items). BSTMap is slightly slower than the ladder two because it has to iterate through a log n amount of positions to get to the end to insert the new node.



Hashmap and arraymap are both slow because they must iterate through all values and then sort through the list after. Binsearch map is fastest because all it has to do is iterate through the list from start to finish. BSTMap is a bit slower than binsearch because it goes to the nodes out of order.

	ArrayMap	LinkedMap	BSMap	HashMap	BSTMap
Insert	O(1)	O(1)	O(n)	O(n)	O(n)
Erase	O(n)	O(1)	O(n)	O(n)	O(n)
Contains	O(n)	O(n)	O(n)	O(n)	O(n)
Find_Keys	O(n)	O(n)	O(n)	O(n)	O(n)
Sorted_Keys	O(n logn)	O(n logn)	O(logn)	O(n logn)	O(n)

Challenges:
This project has been my smoothest so far. Despite being discouraged by erase after a few failed attempts, I eventually got it working and it was mainly smooth sailing from there. I got the rest of the functions done much quicker than usual. I think I intuitively understand how to write the recursive functions despite not being great at visualizing the processes in my head. When I did have problems in my code, I seem to be much better at troubleshooting and debugging those errors than earlier in the semester. Overall, my biggest hiccups on the project was the erase function acting oddly and me accidentally forgetting to set the right pointer to nullptr on the copy assignment function.