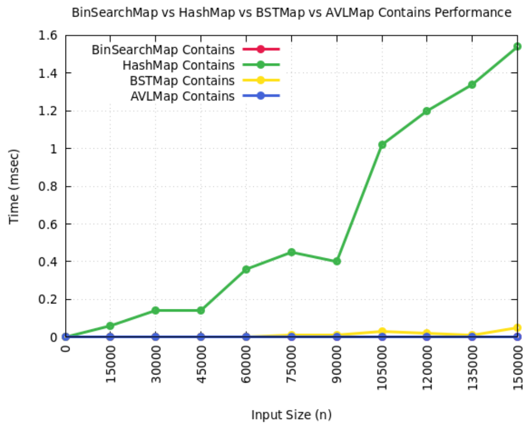
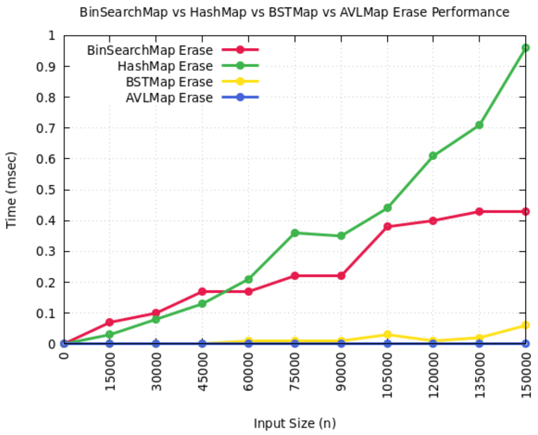
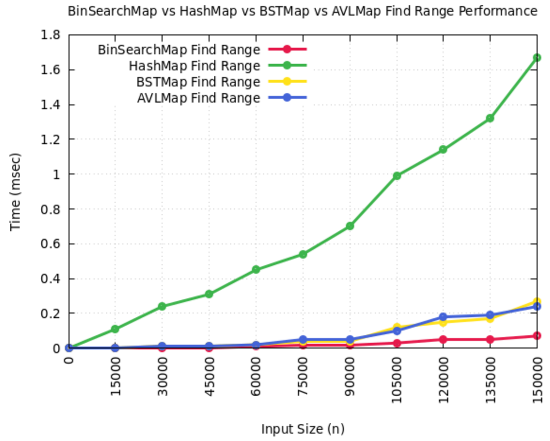
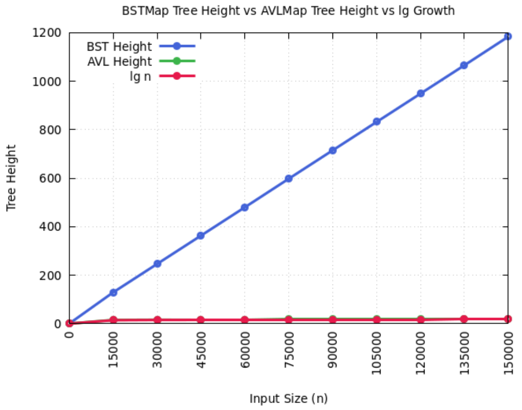
Colleen Lemak
Professor Bowers
CPSC 223
10 December 2021

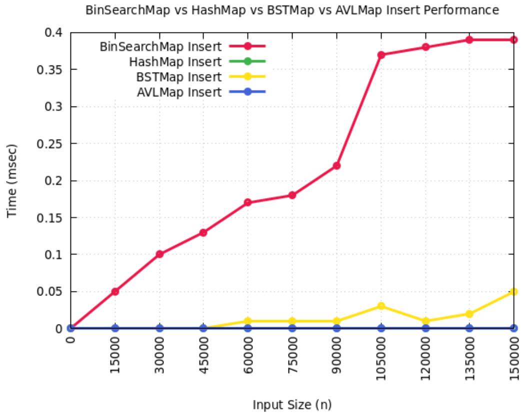
hw8_write_up

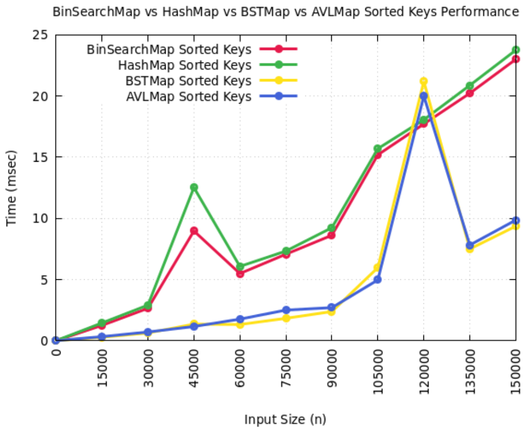












Explanations:
I found AVLMap interesting to implement because the heights of the nodes are stored in the nodes themselves. This made it slightly tricky to implement, but easier to do certain functions with more complexity. BSTMap and AVLMap are fairly similar because they are able to use recursion to navigate to the proper place for the respective function. The AVLMap is much faster than BinSearchMap in insert because the AVLMap recursively goes to the node's position and is able to backtrack and update heights along the way, whereas BinSearchMap needs a more iterative approach which is much more expensive. Another important difference is that AVLMap navigates a path muchlike BSTMap, and so the cost of doing a function like contains is very minimal, as opposed to another structure such as HashMap which needs to check every node for the key, which is more time consuming.

Operation	ArrayMap	LinkedMap	BinSearch	HashMap	BSTMap	AVLMap
insert	O(1)	O(1)	O(n)	O(1)	O(n)	O(logn)
erase	O(n)	O(n^2)	O(n)	O(1)	O(n)	O(logn)
contains	O(n)	O(n^2)	O(logn)	O(n)	O(logn)	O(logn)
find_keys	O(n)	O(n^2)	O(n)	O(n)	O(n)	O(n)
sorted_keys	O(n^2)	O(nlogn)	O(n)	O(nlogn)	O(n)	O(n)

Challenges:
Updating the heights of the graphs was tricky; I found it challenging to implement some of the rotations but drawing everything helped with the various cases.