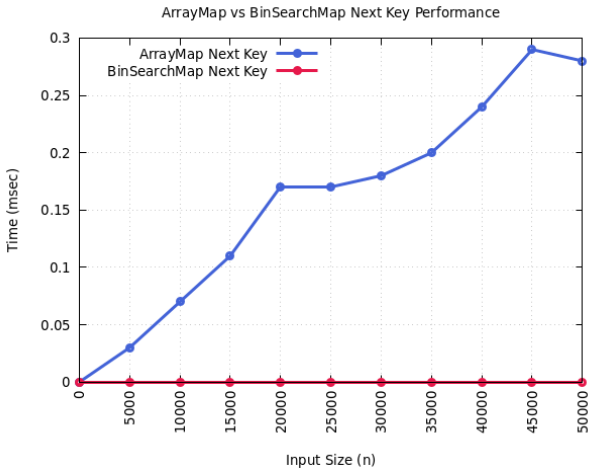
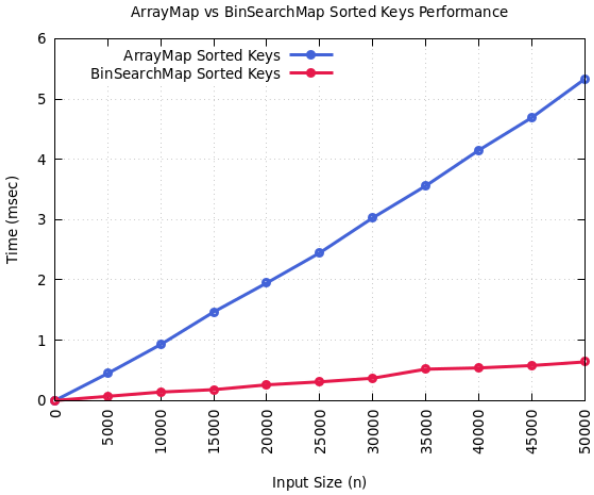
HW6 Writeup (Spring 2022)

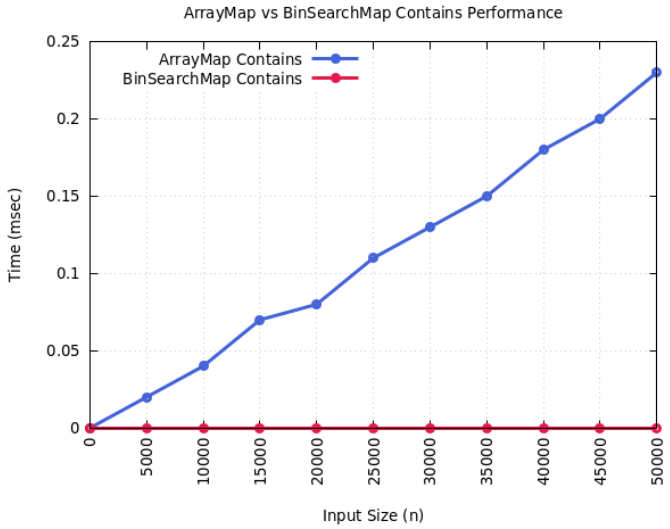
Plots



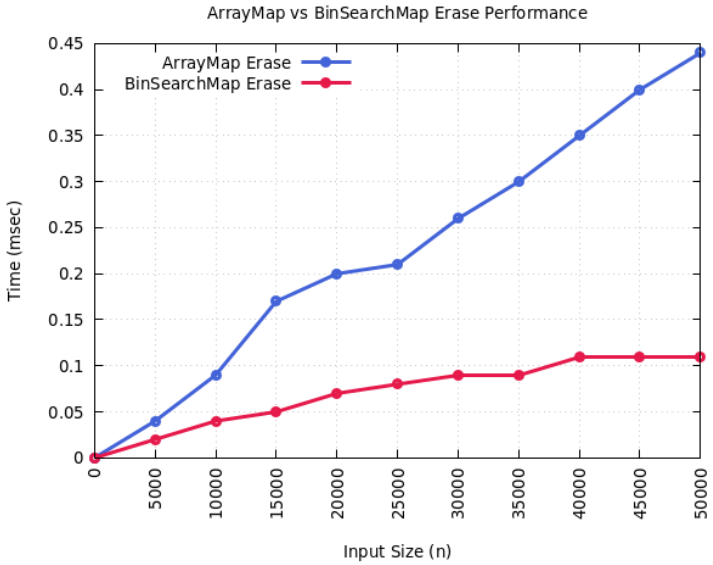
The plot above, a next_key() performance comparison between my ArrayMap and BinSearchMap implementations is shown above. Due to next key in Binary Search Map calling the bin_search function to find a following key rather than iterating through the list, BinSearchMap performs better when talking about time complexity.



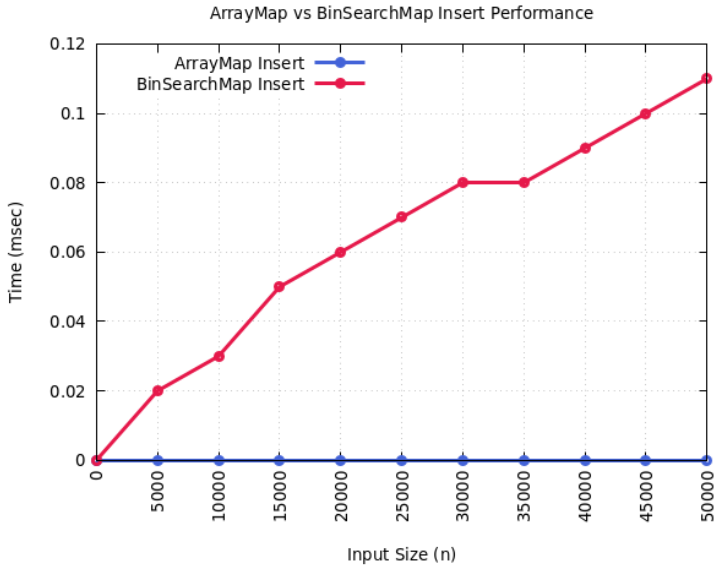
This plot shows the performance comparison of the sorted_keys() function. In ArrayMap, this function requires a call to a sorting function housed as a method of the ArrayMap class to sort the keys. In contrast, BinSearchMap maintains a sorted list of items thanks to how the insert function is implemented. Thus, it makes sense that BinSearchMap would be faster in this comparison.



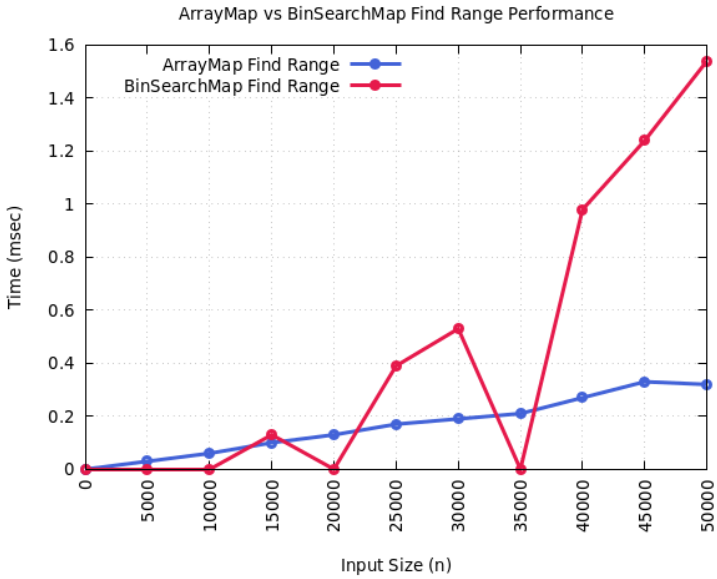
The contains() plot shows a stark contrast between ArrayMap and BinSearchMap. The contains function for ArrayMap needs to iterate to check for the index, while BinSearchMap only requires a call to bin_search which is quicker than looping through the sequence. This is why the graph above shows a linear time complexity for ArrayMap and what appears to be constant time for BinSearchMap.



The erase function, a method of the map interface, has been implemented differently between BinSearchMap and ArrayMap. While the BinSearchMap erase only calls bin_search to find a proper index to erase, ArrayMap makes a call to contains and then still must iterate through the sequence to find the proper key to remove at. Still, both appear to be relatively quick in terms of time complexity.



This plot differs from the previous ones as it shows a worse overall time complexity in the BinSearchMap. This is correct as the sacrifice with BinSearchMap involves a slower insert method due to the need for checking where a key-value pair should be inserted at rather than just inserting at the end of the sequence. Therefore, it should be of no surprise that this graph looks in opposition to the others.



Finally, the find_keys() performance graph displays some even more interesting behavior. While the ArrayMap version stays linear in time complexity, the BinSearchMap implementation jumps back and forth across the ArrayMap line. This can be attributed to the bin_search call in find_keys() for the BinSearchMap function where there is a more selective process for copying just the keys from the sequence.

Challenges

I faced large problems with comprehending how to utilize the bin_search function to implement key-related functions: find, next, and previous key. These gave me the most trouble because, though next and prev are very similar, they still required a nested if statement just like find keys. The trouble was specific to the correct ordering of assignment operations as well as proper consideration of all necessary conditional checks.