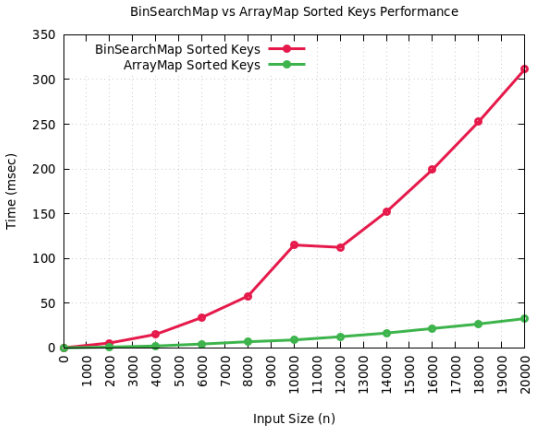
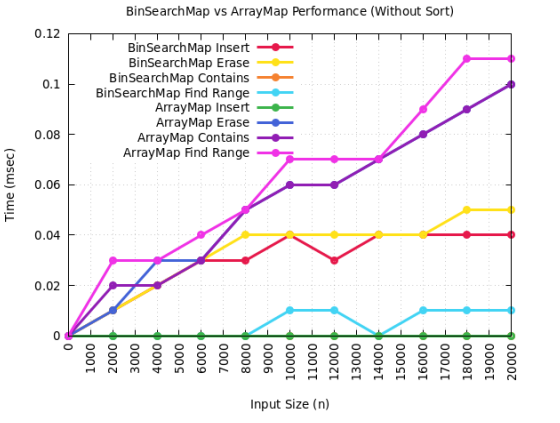
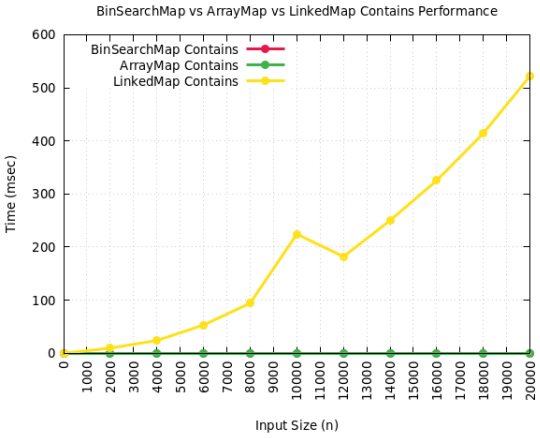
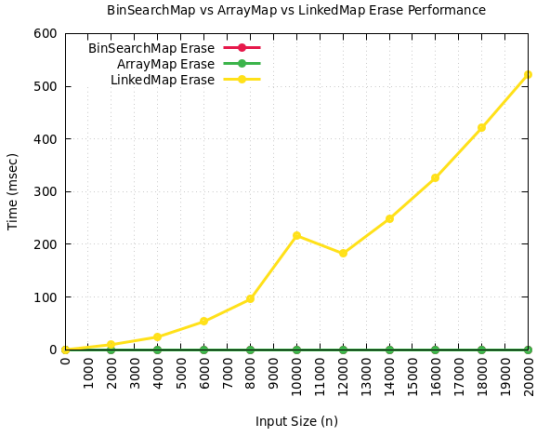
Colleen Lemak
Professor Bowers
CPSC 223
28 October 2021

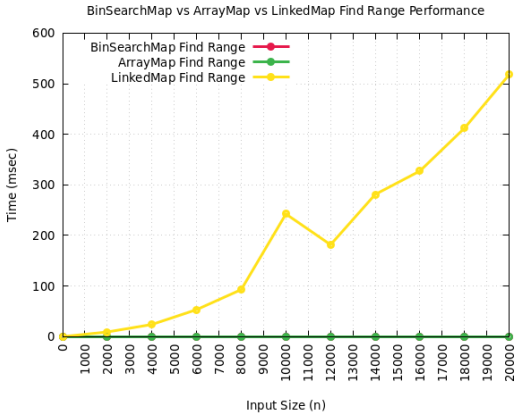
Hw5_write_up

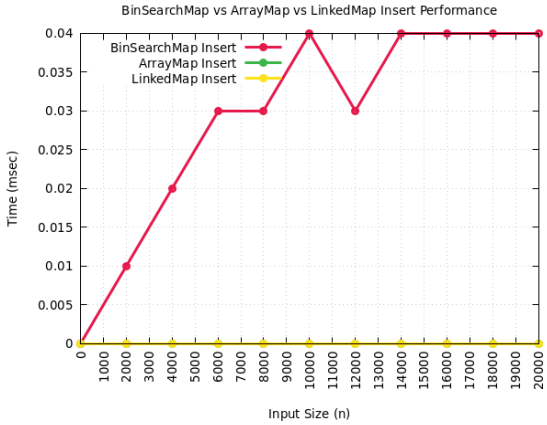


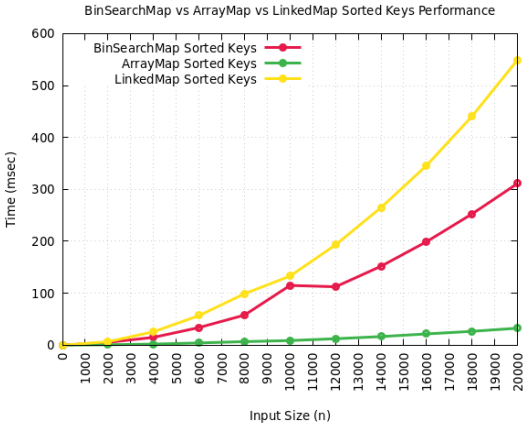












These performance graphs support our whole purpose of using BinSearch() to speed up various tasks. Using this helper function within other functions allowed us to directly index to where we needed to, without looping through and tracking down a position like ArrayMap and LinkedMap. For example, when calling find_keys(), you would normally have to index all the way to k1 and k2 then use loops, but with bin_search(), it is much easier to find the keys that fit the condition because of our helper function's implementation. Typically these maps were slower than BinSearchMap because of the loops needed, however, BinSearchMap is slower in the insert function, as expected and shown on the insert graph in red. Additionally, the sorted_keys() BinSearchMap appeared to run faster than the LinkedMap, but slower than the ArrayMap in green because of an array's ability to quickly use sorting algorithms.

Using bin_search() in BinSearchMap was tricky for me, but the other two maps weren't too complex, I just had to get used to the new key pair value syntax.

Map Implementation

Operation	ArrayMap	LinkedMap	BinSearchMap
insert	O(1)	O(n)	O(logn)
erase	O(n)	O(n)	O(n)
contains	O(n)	O(n)	O(logn)
find_keys	O(n)	O(n)	O(logn)
sorted_keys	O(n^2)	O(nlogn)	O(n)