

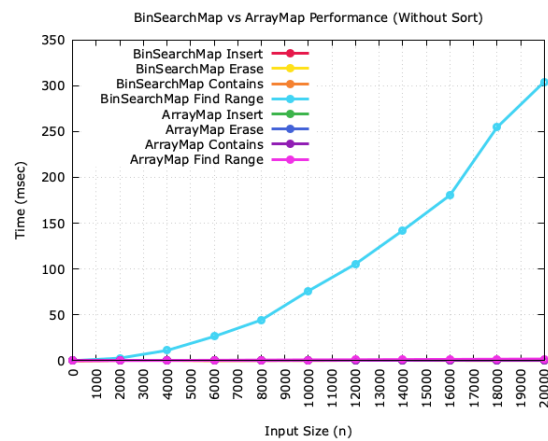
NAME: Ben Puryear

FILE: HW-5_WriteUp.pdf

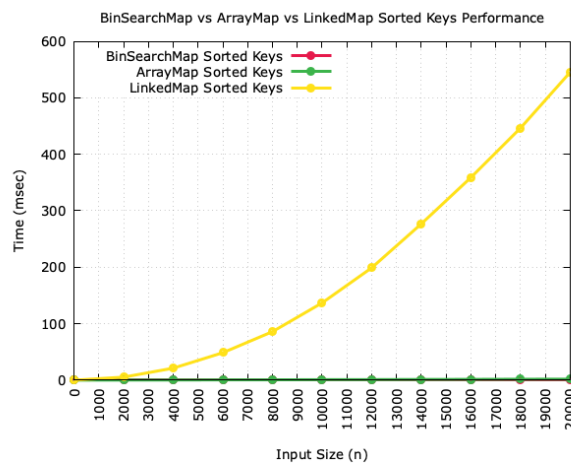
DATE: Fall 2021

DESC: This pdf goes over the basics accomplished throughout all the HW-4 related files.

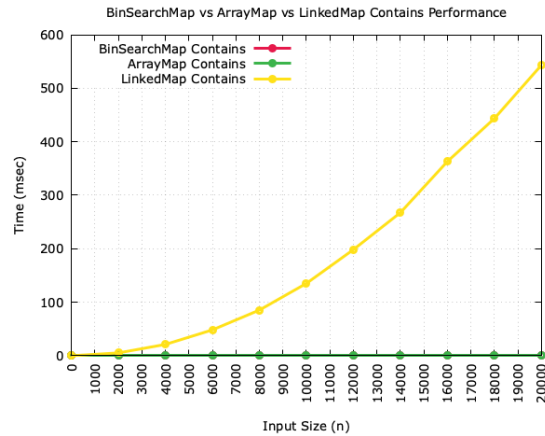
Graphs / Explanations



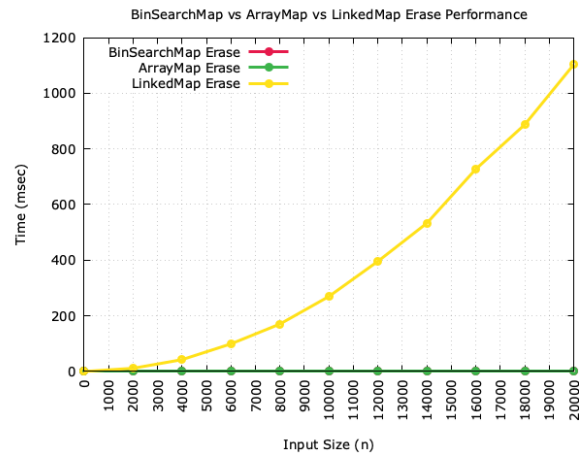
This graph displays the general performance metrics of the three different maps in HW5. One thing that stands out is ArrayMap Find Range. This



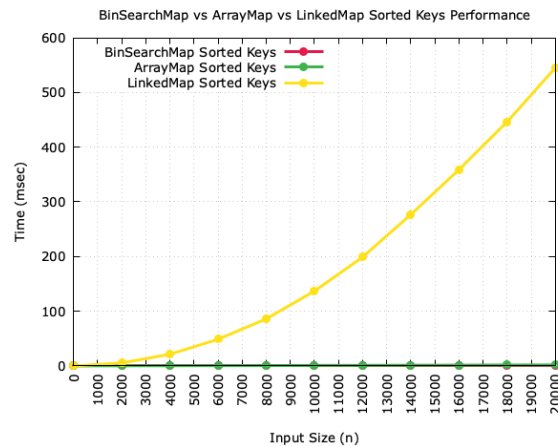
This graph shows the performance metrics of BinSearchMaps and ArrayMaps sorted keys. The reason that Binsearchmap is so much faster is because it is already sorted.



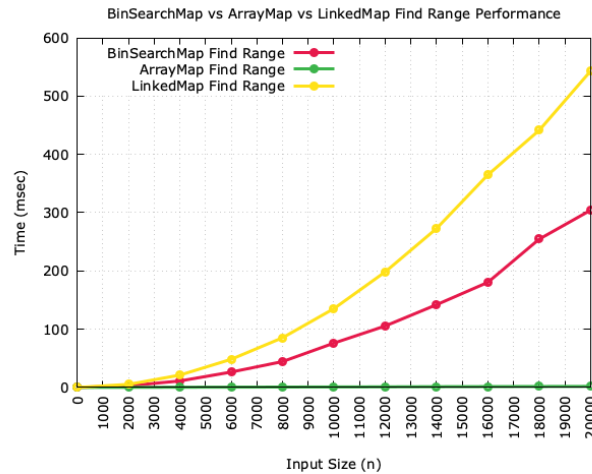
This graph shows the different implementations of Contains. My theory on why LinkedMap is so much worse than the other two is because it needs to traverse the array.



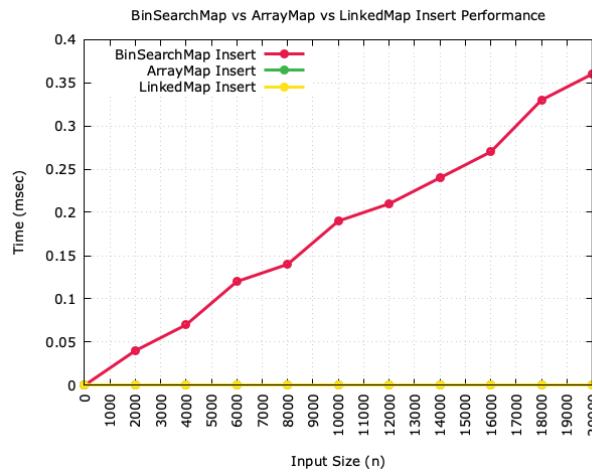
This graph shows the different implementations of erase. My theory on why LinkedMap is so much worse than the other two is because it needs to traverse the array to find the index it needs to erase.



This graph shows the comparison between the different implementations of SortedKeys. I have no idea why LinkedHashMap is so much worse than the others.



This shows the comparison between the different implementations of find range. My theory on why LinkedHashMap has a worse performance than the others is because LinkedHashMap needs to traverse the entire map whereas both array and binsearch don't need to.



This graph shows the comparison between the different implementations of insert. My theory on why BinSearchMap insert takes so much longer than the rest is because both ArrayMap and LinkedHashMap do not require the insertion of the value to be in order, whereas binsearchmap requires the value to be inserted in a sorted position.

Table:

operation	ArrayMap	LinkedMap	BinSearchMap
insert	1	1	1
erase	N	N ²	1
contains	N	N ²	1

find keys	N	N^2	N
sorted keys	N^2	N^2	N

Issues I faced:

No issues.