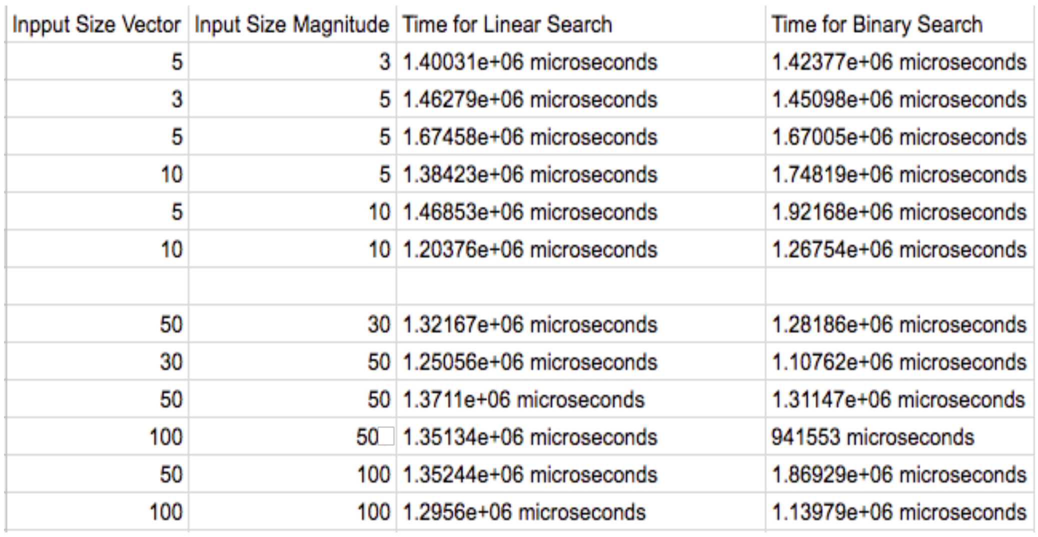
Faustine Yiu
913062973
ECS 36C: Programming 1 Report

Linear Search vs. Binary Search CPU Time Analysis

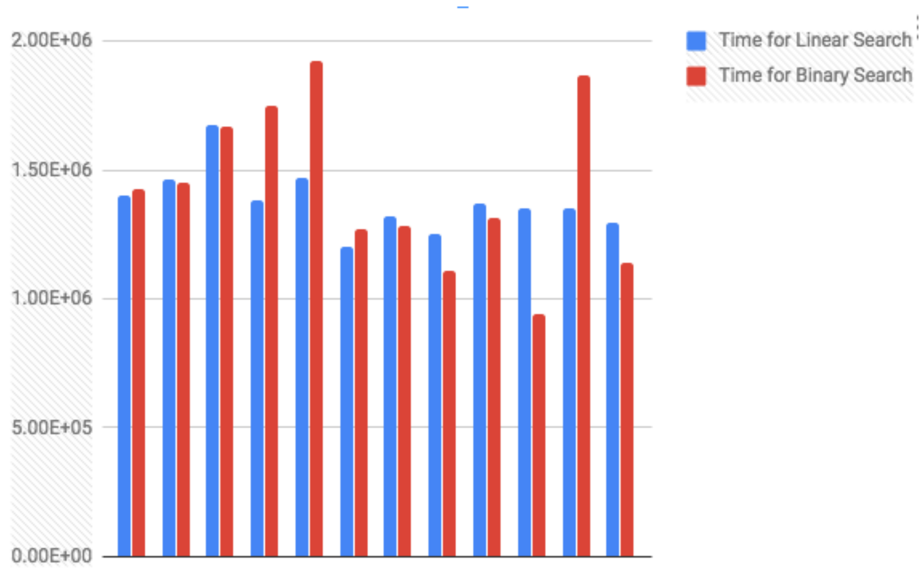
This is an analysis of the CPU efficiency of the linear and binary search for the programming assignment of searching for a match between vectors and a range of given magnitudes for scientific studies. All following computations are taken from the folder: /home/cs36c/public/p1/vector_search_reference from CSIF

Below is a graph that shows the number of runtimes, correlated with its input size and duration. As we may observe, there is no obvious correlation between stating whether linear search is more favorable over binary search. And thus, we need to break this graph down into plots to understand the numbers in more correlation.



Inpput Size Vector	Input Size Magnitude	Time for Linear Search	Time for Binary Search
5	3	1.40031e+06 microseconds	1.42377e+06 microseconds
3	5	1.46279e+06 microseconds	1.45098e+06 microseconds
5	5	1.67458e+06 microseconds	1.67005e+06 microseconds
10	5	1.38423e+06 microseconds	1.74819e+06 microseconds
5	10	1.46853e+06 microseconds	1.92168e+06 microseconds
10	10	1.20376e+06 microseconds	1.26754e+06 microseconds
50	30	1.32167e+06 microseconds	1.28186e+06 microseconds
30	50	1.25056e+06 microseconds	1.10762e+06 microseconds
50	50	1.3711e+06 microseconds	1.31147e+06 microseconds
100	50	1.35134e+06 microseconds	941553 microseconds
50	100	1.35244e+06 microseconds	1.86929e+06 microseconds
100	100	1.2956e+06 microseconds	1.13979e+06 microseconds

In the graph to the right, we see that there are only some instances where binary search is barely faster than linear search, the most by ~ 50E+06. However, there are multiple instances where linear search is clearly faster than binary search, the most also ~ 0.5E+06.



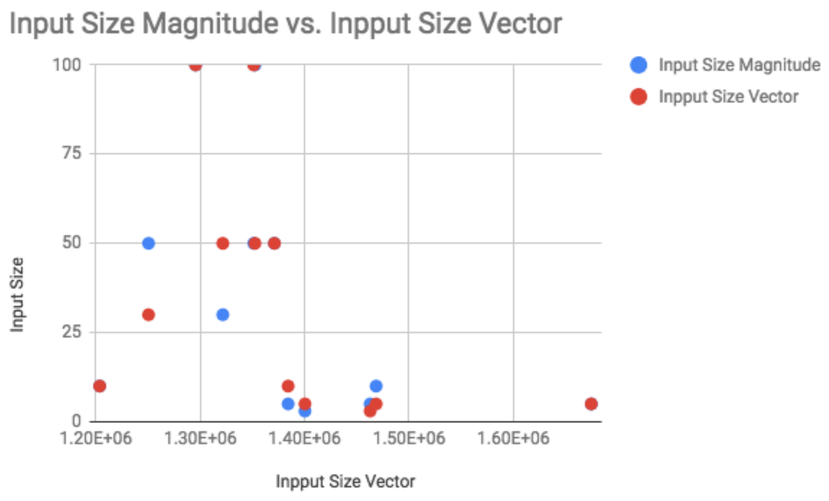
The observation with linear search is that it is more or less consistent, or at least more consistent when compared to binary search, while binary search can either be more efficient or less efficient, and thus less consistent.

The lowest amount of computation time, however, results in a binary search, where the vector has 100 inputs and magnitude has 50 inputs, generating at a CPU of 941553 microseconds.

The highest amount of computation time, nonetheless, results also in binary search, where the vector has 5 inputs and magnitude has 10 inputs. Thus, we could conclude that binary search is better/more efficient with larger files.

Overall, there is a correlation between linear search being more efficient for smaller inputs and binary search being more efficient for larger inputs.

Below is another graph that attempts to track the correlation between input sizes. However, as we can see, no matter how small or how big the input size, there is no direct correlation between size of input and time. This is more a less a scatter plot that has no direct correlations. The linear and binary sort returns either amounts of inputs more or less with the same approximate time of between 1.20E+06 and 1.60E+06.



In conclusion, as we calculate the big O notation of the linear and binary sort, we get O(n) for linear and O(log n) for binary. It is shown with the big O notation that binary search has a more efficient order of growth computation than linear. However, through our analysis, this does not mean that it is always better to use binary search over linear search. It also depends on the number of input and where the "match" or value you are looking for is in the vector/array. And thus, for our programming assignment, we would suggest the scientists to conduct binary searches when they have a larger input and do not really understand how the numbers are structured in memory, versus using linear if it is a smaller input and they understand that the matches are in the beginning or the memory.