Raymond Chan
10/08/18
ECS 36c
Joël Porquet, PhD

Measurement Analysis between Linear Method and Binary Method

Purpose:
This program's purpose is to evaluate the pros and cons of linear search and binary search, given different input sizes.

Scenario:
The program will receive different input sizes, in the form of two type of inputs: positional vectors and magnitudes. Given any number of positional vectors and magnitudes, the program is to find if the any given magnitudes matches the modulus (aka the magnitudes) of the positional vectors.
The program is to search for appropriate matches in one of two ways: linear search or binary search. It is up to the user to choose the method of searching, as well as the amount of data to input for the program to process.
At the end, the program will display the amount of matches found between the two input, along with the total CPU time for completing the search in microseconds. The amount of matches found should be written in a res_n.dat file.

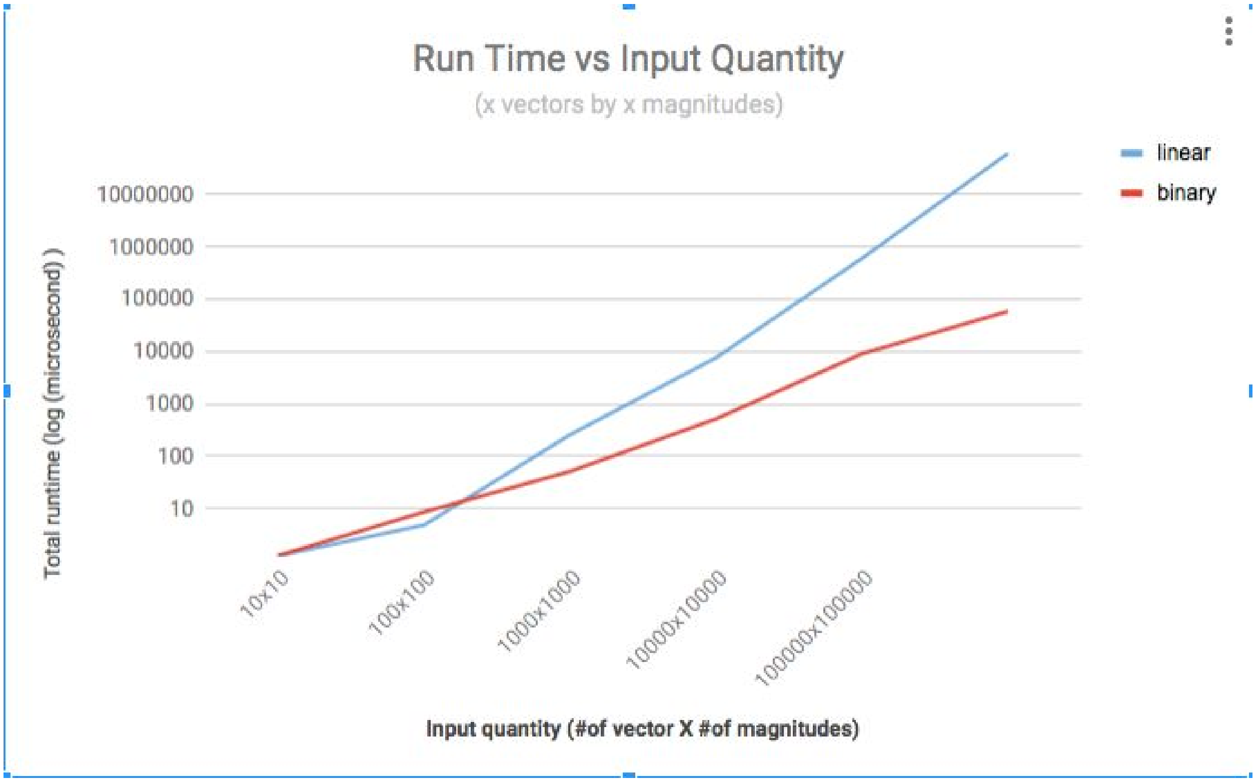
Methods:
Up to 100,000 magnitude inputs and 100,000 positional vector inputs were processed by the program at once. Each runs have varies matches of inputs between vector and magnitude; each inputs have the starting amount of one input, and then later increase its amount of input by the factor of 10, as an example:

Number of positional vector input	Number of magnitude input
1	1
1	10
...	...

1	100,000
10	1
10	10
...	...
100,000	100,000

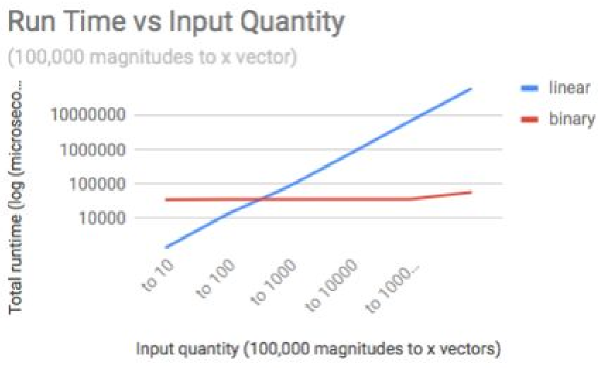
Each matches are computed by the program at least 3 times to minimize the margin of error due to inconsistency in both program run time and the internal stopwatch. The average of each run will be recorded in order to evaluate the relationship between search methods, run time, and input sizes.

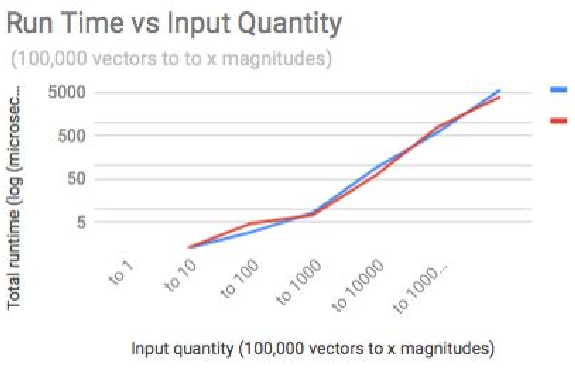
Result:
When comparing linear and binary search with x vectors and x magnitudes of inputs:



Although linear search method has the advantage when processing below 100x100 input size, it run time is clearly growing exponentially (i.e. T(N) = O(X^N)) as input size become more and more ginormous, while the binary search method only has a logarithmic growth (i.e. T(N) = O(log(N))(be mindful that this graph present run time in log scale)

Below are representation of run time vs input quantity relationship, where the left graph has constant magnitude input of 100,000, and the right graph has constant vector input of same amount:





It is clear that binary search is best at dealing with huge amount of magnitudes input with the growing amount of vectors input (shown on the left), while large quantity of vector inputs result in both linear and binary search method having comparable run time.

Conclusion:

Based on the data from above, we can conclude that binary search is better at handling large amount of data, while the linear search is only getting slower and slower as the input size increase, regardless of with input size in larger.

However, linear search method is most ideal if the input size is less than 100 magnitudes to 100 vectors, since the growth rate of linear has not surpass the growth rate of binary before that.

Interestingly, having a large vector input size ensure both search method's growth rate matches with each other. This might have to do with binary search needed to organize the whole vector inputs before getting to the searching part. In other word, we can say that the Big O of binary search for sorting is comparable to the Big O of linear method's actual search time.