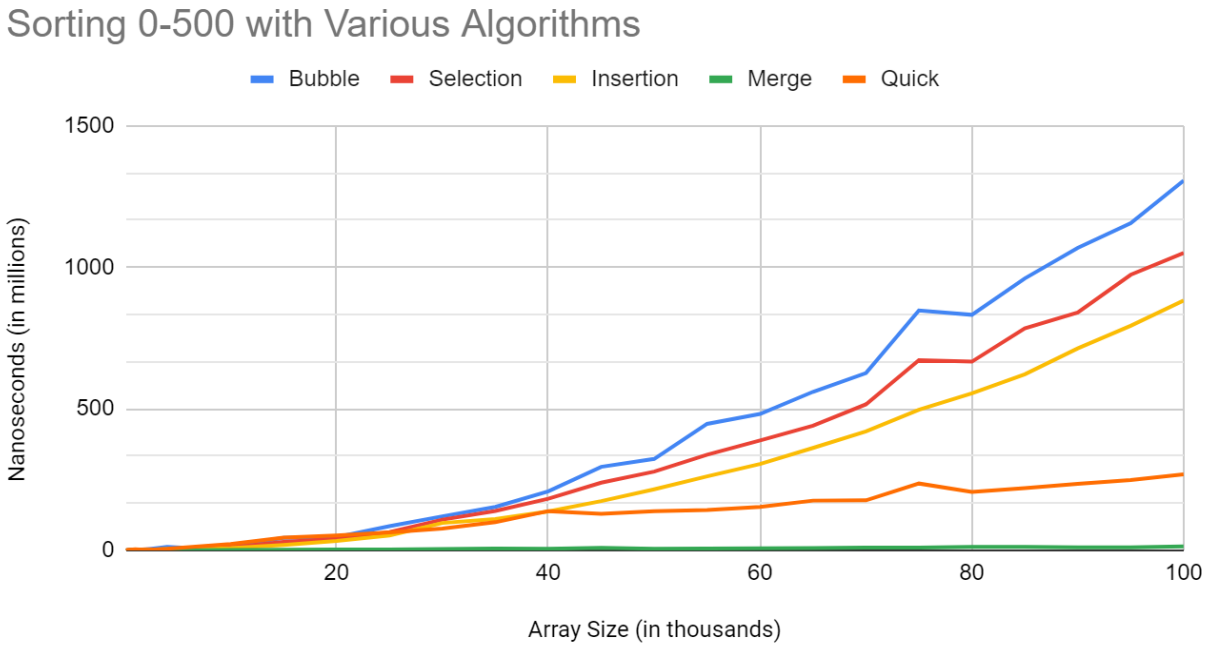
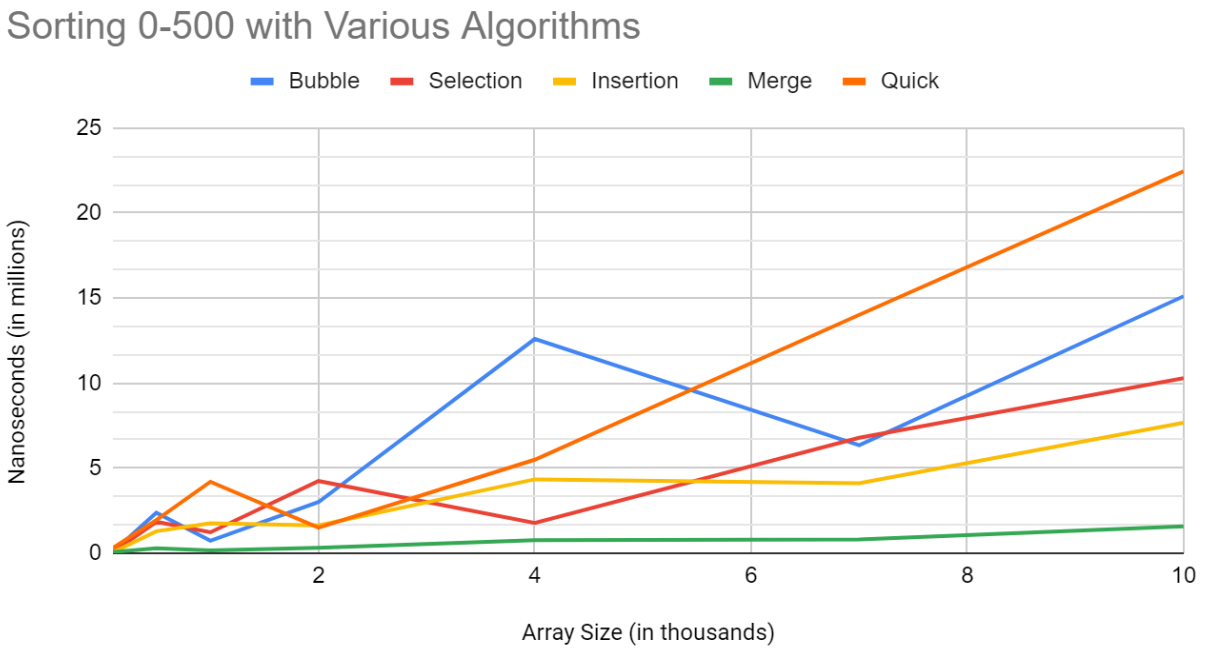

Sorting 0-500 with Various Algorithms
Bubble
Selection
Insertion
Merge
Quick
1500
1000
500
0
Nanoseconds (in millions)
20
40
60
80
100
Array Size (in thousands)


Sorting 0-500 with Various Algorithms
Bubble
Selection
Insertion
Merge
Quick
25
20
15
10
5
0
Nanoseconds (in millions)
2
4
6
8
10
Array Size (in thousands)

Looking at similar runtimes, Bubble, Selection, and Insertion sorts appear to have similar runtime curves, increasing exponentially as the size of the array increases.

Merge and Quick sorts seem to be similar in the fact that their slopes are fairly linear.

Comparing this analysis against their complexities, Bubble, Selection, and Insertion sorts having a complexity of O(n^2), this aligns exactly with what is observed in the graph– an exponential increase in runtime.
Quick and Merge Sorts have a complexity of O(nlogn), which easily explains the dramatic difference in runtime curves, which appear nearly linear.

Ranking the sorting methods based on their final results, Bubble is the slowest, then Selection, Insertion, Quick, and the fastest, Merge sort.

A final, general observation is that the runtimes have an irregular increase at an array size of 75,000, for either a processing reason or something in how arrays are sorted.