Laboratory VII

Benjamin Streit

EECS 268

October 30, 2017

I. **Do the sorts we know to be of O(n^2) complexity demonstrate this behavior?**

Yes, these algorithms demonstrated fairly consistent completion times with arrays of increasing sizes. Sorting algorithms of O(n^2) can be ideal for smaller data sets, but decrease in efficiency with larger data sets. This was experienced with the bubble sort, selection sort, and insertion sort algorithms as they performed on par or better with the algorithms of O(n*log(n)) for small data sets, but were unable to compete when approaching the larger data sets of the tests.

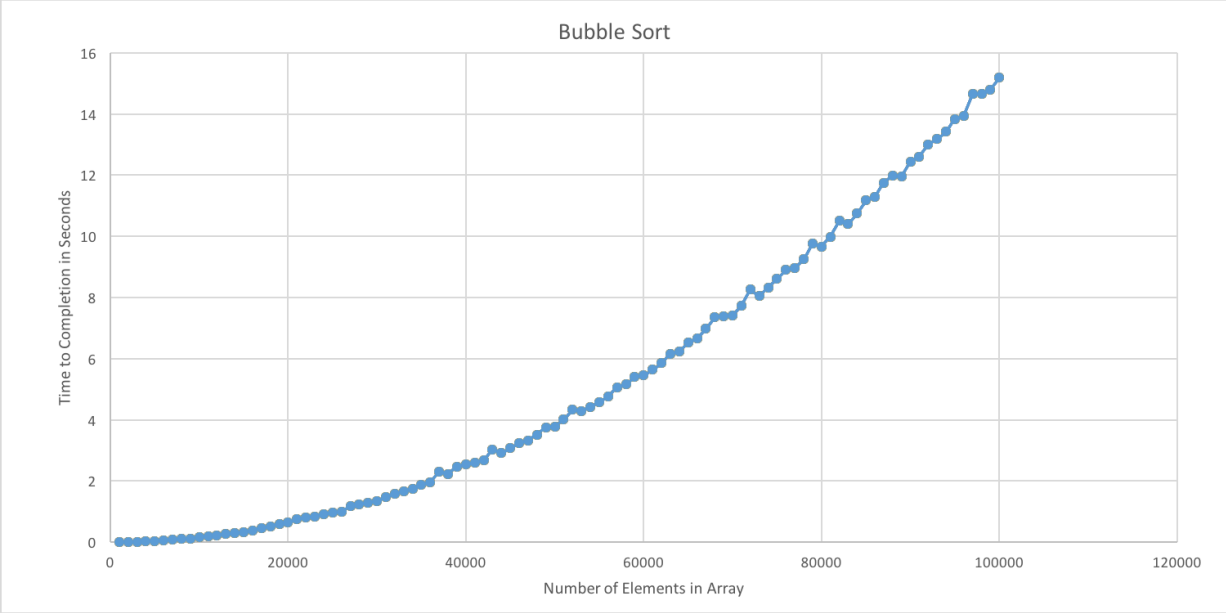
II. **Do the sorts we know to be of O(n*log(n)) complexity demonstrate this behavior?**

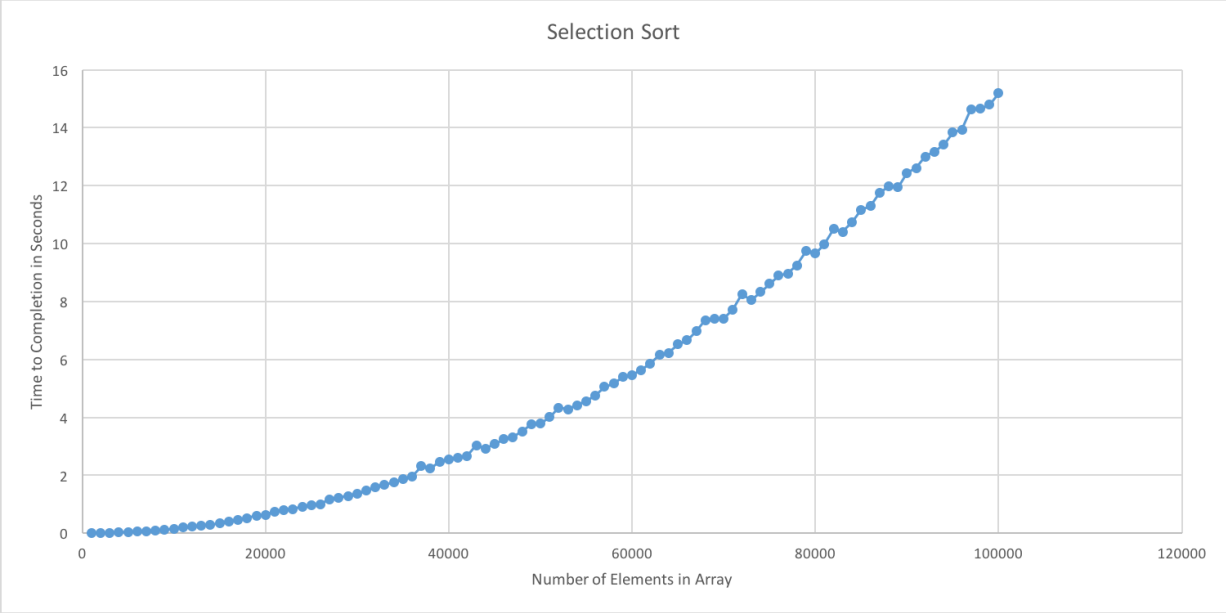
Yes, the algorithms of O(n*log(n)) demonstrated expected behavior of performing well with large data sets. The recursive algorithms of merge sort, quick sort, and quick sort with median were able to complete sorts with increasingly larger arrays than the non-recursive algorithms of O(n^2) complexity.

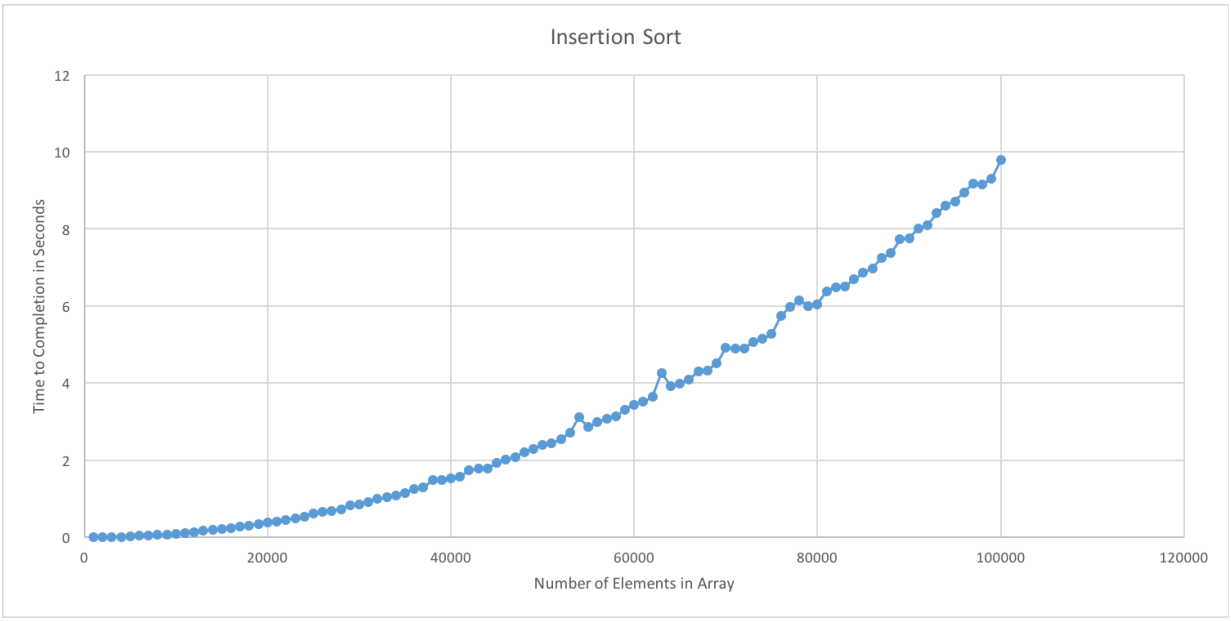
III. **For any plot points that are outliers, how do you explain them?**

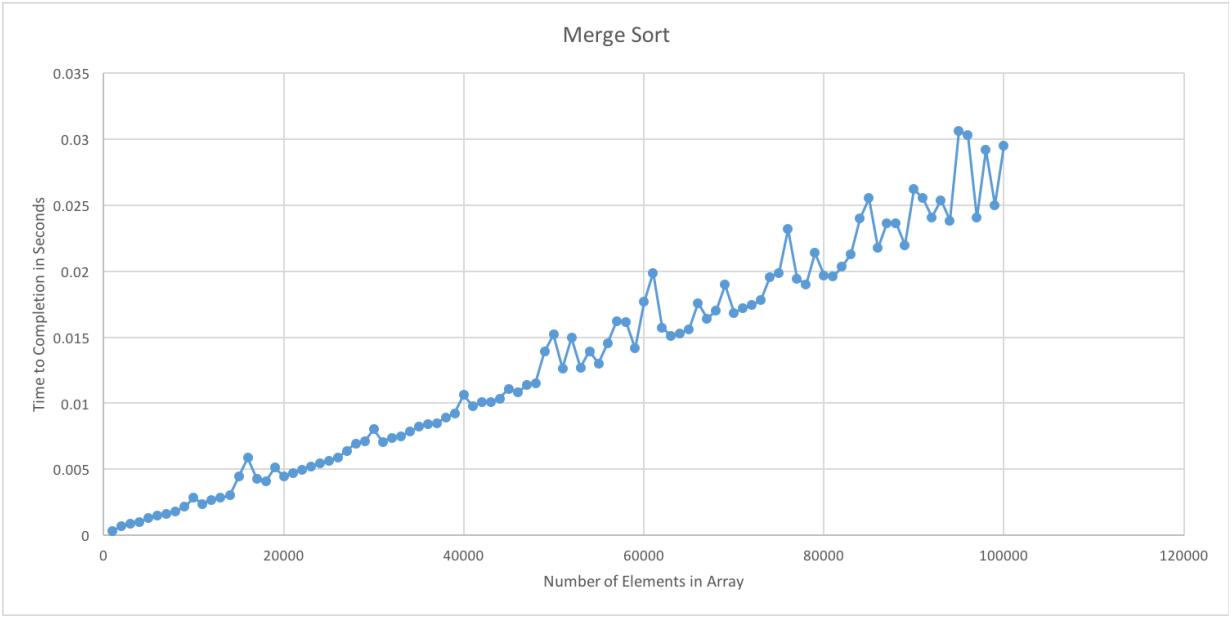
Several of these algorithms have best and worst case scenarios, an outlying point on a plot is an example of one of these best or worst case scenarios occurring.

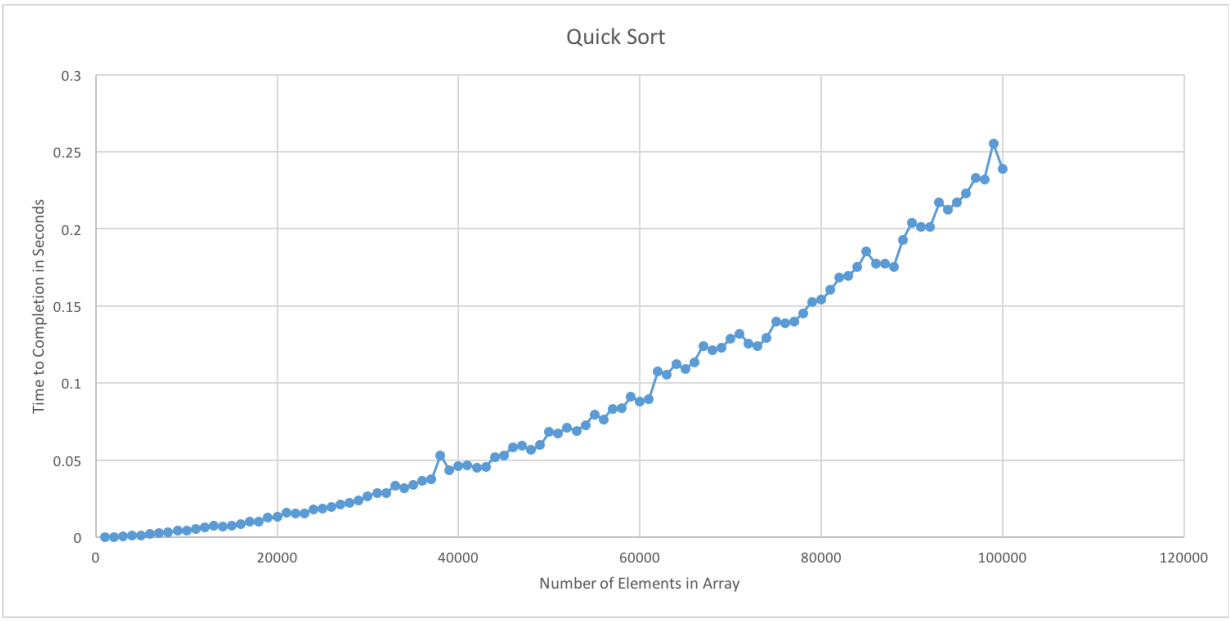
IV. Data Plots

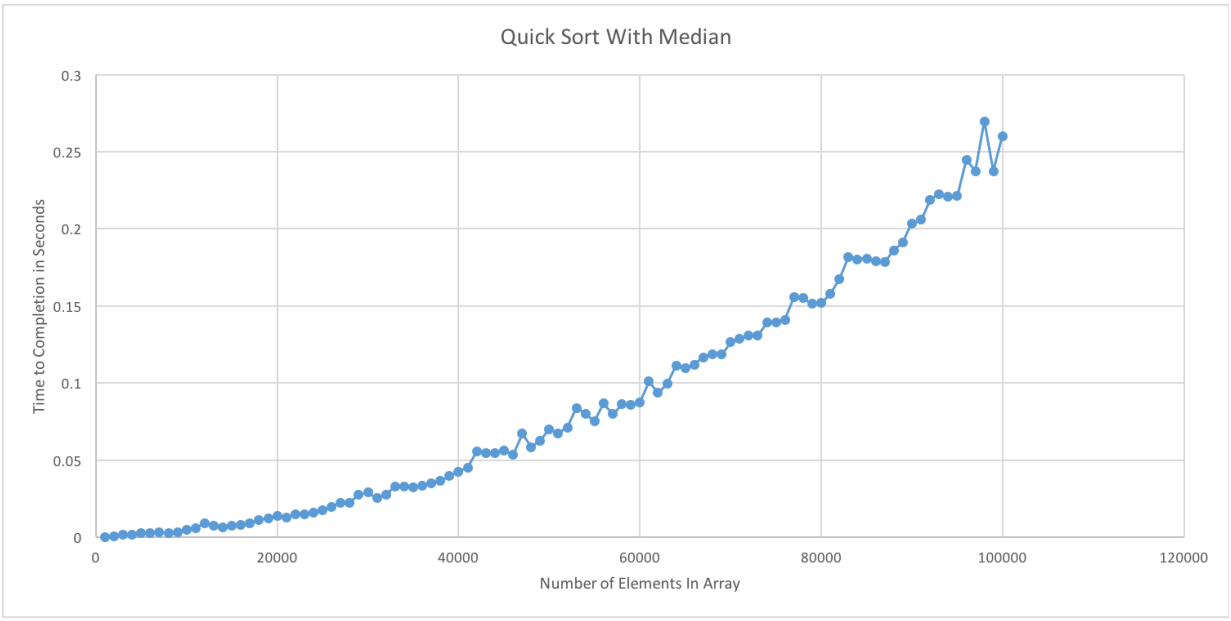





Insertion Sort
Time to Completion in Seconds
Number of Elements in Array
0
2
4
6
8
10
12
0
20000
40000
60000
80000
100000
120000


Merge Sort
Time to Completion in Seconds
Number of Elements in Array
0
0.005
0.01
0.015
0.02
0.025
0.03
0.035
0
20000
40000
60000
80000
100000
120000


Quick Sort
Time to Completion in Seconds
0.3
0.25
0.2
0.15
0.1
0.05
0
0
20000
40000
60000
80000
100000
120000
Number of Elements in Array


Quick Sort With Median
Time to Completion in Seconds
0.3
0.25
0.2
0.15
0.1
0.05
0
0
20000
40000
60000
80000
100000
120000
Number of Elements In Array