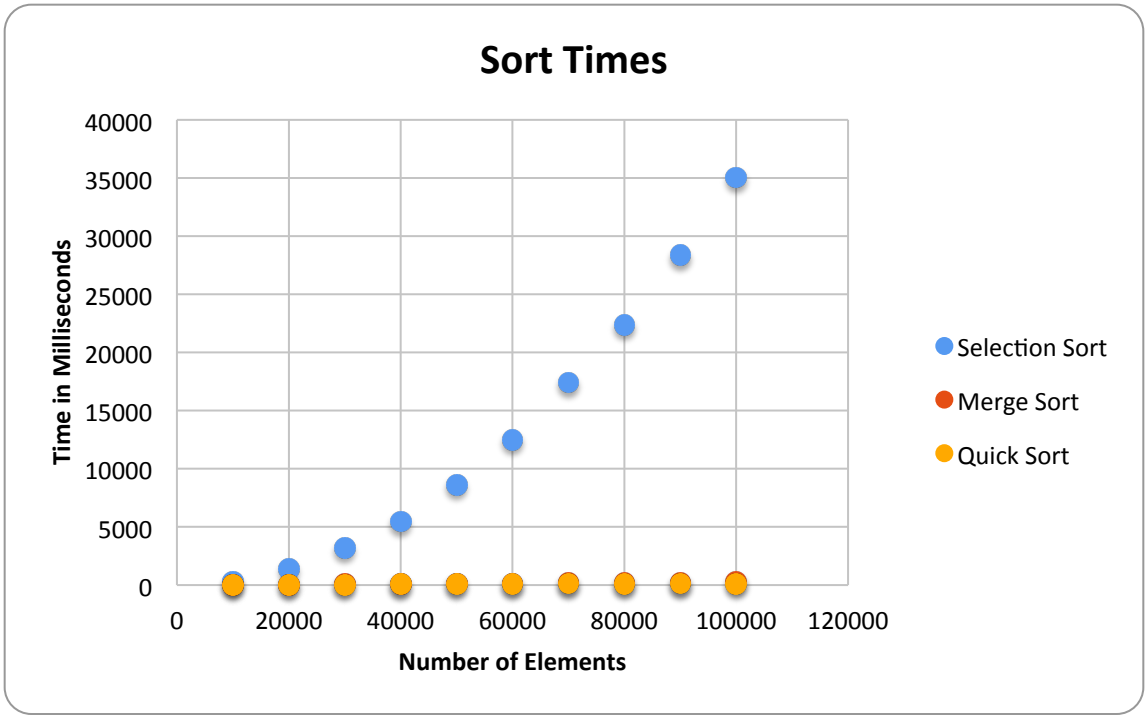
Report

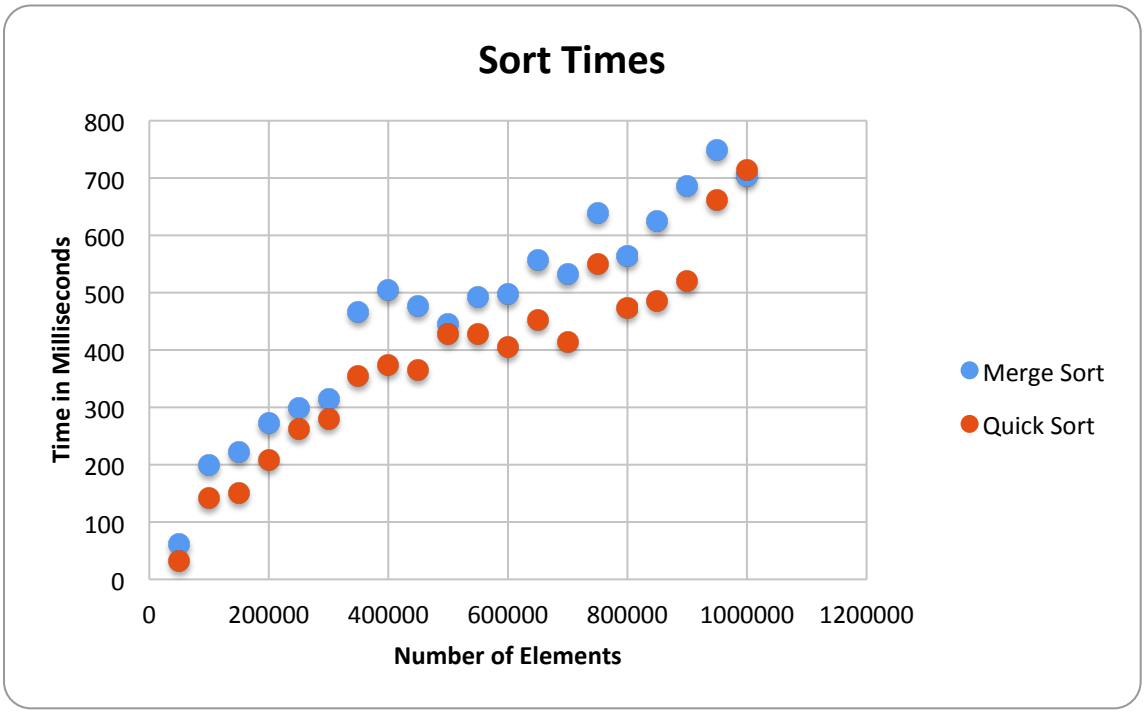
The data shows that the selection sort followed the $O(n^2)$ idea, meaning that it was an exponential curve that increased dramatically as n increased, compared to both the merge and quicksort, which hovered together around O(nlogn) curve.

This data was basically expected, and it should be noted that the quicksort did not run into the trouble of duplicates that would have caused it to behave more like O(n^2), since it was using an array of Doubles with more than 6 significant digits.

Num. of Elements	Selection Sort	Merge Sort	Quick Sort
10000	314.094061	9.181328	11.922147
20000	1393.43326	43.047849	22.238886
30000	3172.256075	81.808419	35.611574
40000	5469.284834	113.445757	52.767559
50000	8619.656333	134.523049	107.128496
60000	12489.11402	116.186496	85.44264
70000	17408.37829	228.33904	128.024839
80000	22340.13199	198.629232	87.991688
90000	28351.46048	207.649007	115.458673
100000	35049.51148	268.340134	95.386659

Num. of Elements	Merge Sort	Quick Sort
50000	61.359108	32.679686
100000	199.370853	141.306451
150000	222.730857	150.752072
200000	272.261722	208.370273
250000	298.015553	261.89709
300000	315.080944	279.890223
350000	466.460636	354.984549
400000	504.258788	374.204664
450000	476.595173	365.031199
500000	445.997376	427.686661
550000	492.426942	427.381854
600000	497.549179	404.883125
650000	556.397884	452.452881
700000	532.517435	414.368394
750000	639.239407	549.950641
800000	563.944213	473.62937
850000	624.630107	485.594718
900000	686.428793	520.065655
950000	749.535652	662.130669
1000000	704.066114	714.314691


Sort Times
Time in Milliseconds
40000
35000
30000
25000
20000
15000
10000
5000
0
0
20000
40000
60000
80000
100000
120000
Number of Elements
Selection Sort
Merge Sort
Quick Sort


Sort Times
Time in Milliseconds
800
700
600
500
400
300
200
100
0
0
200000
400000
600000
800000
1000000
1200000
Number of Elements
Merge Sort
Quick Sort