Megha Madan

Programming Project 3 Report

Initial Results:

The initial results of my three sorting implementations were as expected. Selection sort slowed down much more compared to merge sort and quick sort as the lists began to increase in elements. Merge sort and quick sort were more similar in their run times, and quick sort was slightly faster as it completed the sort inside of the input array without using extra memory space as merge sort does with the temporary array. There was a blip in the data for the merge sort and quick sort between the data points for 40,000 to 60,000 elements, and I am not sure of what caused this. I think it may have been because of something unrelated to the program during the specific run I chose to document.

Improvements:

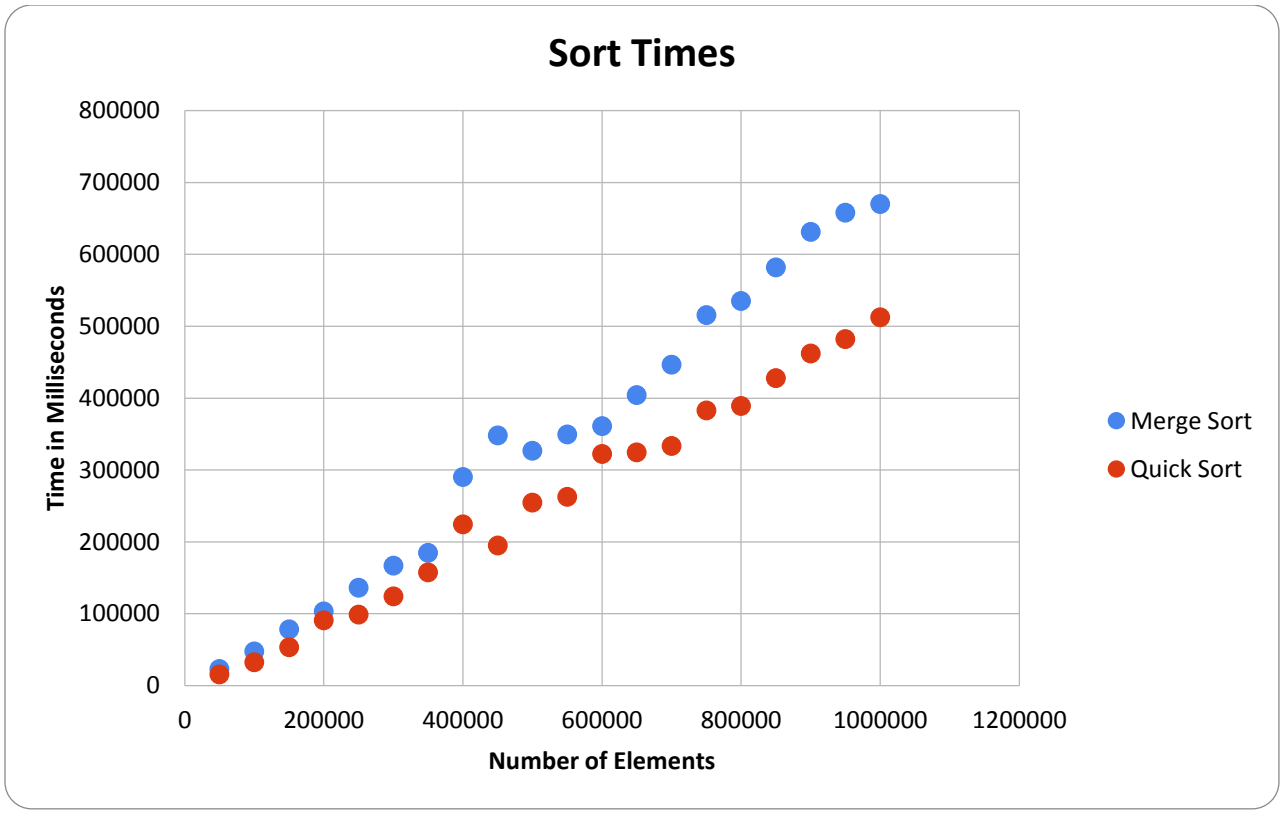
One simple improvement I tried to make is regarding the choice of the pivot in quicksort. Before the improvement, I chose three numbers: the first number, the middle, and the last number and used the median value of these three as the pivot. To make this more efficient, I thought picking random numbers to find the pivot would give me better results. I did not see any improvement, and if there was any change it seemed to only make the numbers a little worse. Then again, picking the median of three numbers seems better than picking the middle or first element of the array each time.

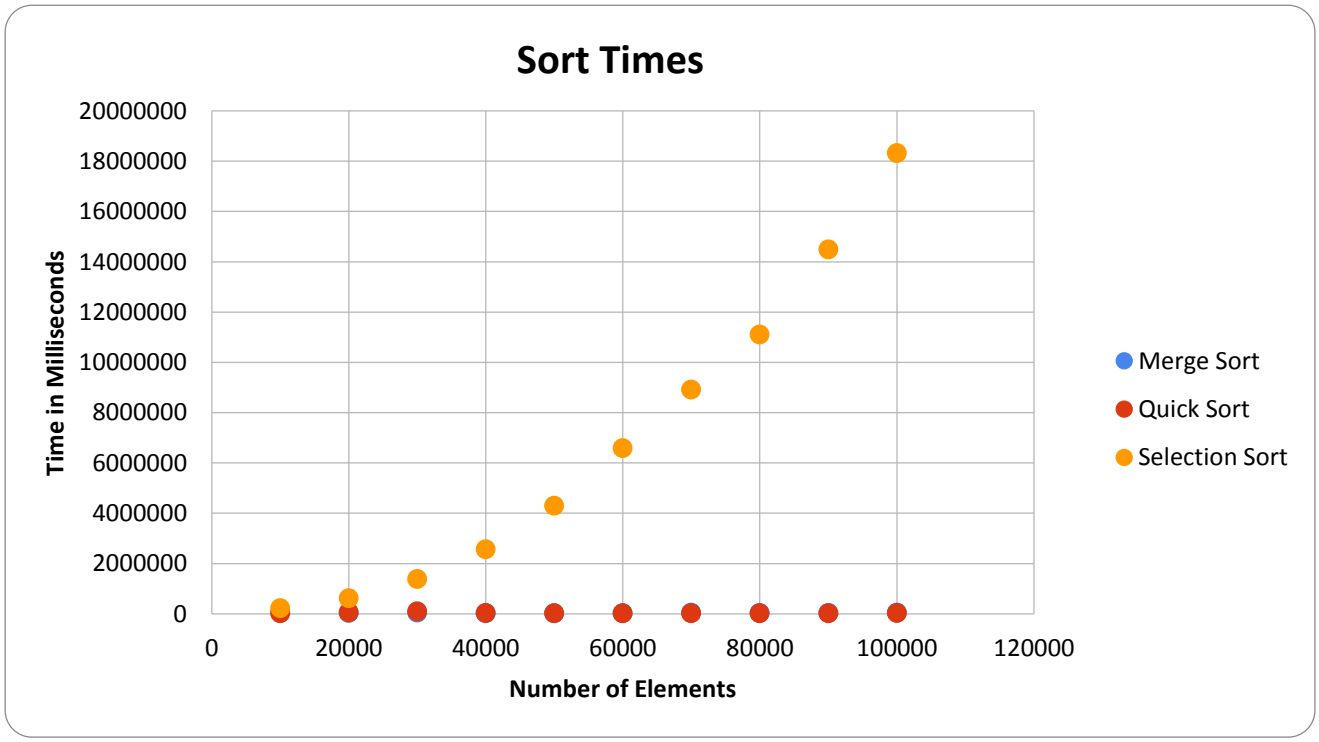
Another improvement to the merge sort and quick sort methods was to switch to selection sort when the number of elements in the array is below a certain threshold (I am using 40 elements as my threshold). I did not see any visible improvement in the results for my quicksort method, and I think this is because insertion sort doesn't have a drastically better performance on these small arrays. Therefore it is difficult to see the improvement on the arrays we sorted.

Results:

Num. of Elements	Merge Sort	Quick Sort	Selection Sort
10000	44183	25941	224560
20000	32016	58572	624211
30000	50620	93625	1380538
40000	18318	29502	2567666
50000	22202	16927	4302230
60000	25876	18458	6590061
70000	41538	20386	8919943
80000	34907	25432	11107268
90000	39185	26573	14491602
100000	44580	33221	18322342

Num. of Elements	Merge Sort	Quick Sort
50000	22955	15572
100000	47543	32384
150000	78332	53375
200000	103255	90808
250000	136134	99026
300000	167096	124086
350000	184849	157597
400000	290294	224605
450000	348100	194969
500000	326727	254894
550000	349688	262519
600000	361194	322404
650000	404075	324454
700000	446798	333567
750000	515462	383081
800000	535162	389269
850000	581800	427893
900000	631322	462354
950000	657874	482121
1000000	670305	512660


Sort Times
800000
700000
600000
500000
400000
300000
200000
100000
0
Time in Milliseconds
0
200000
400000
600000
800000
1000000
1200000
Number of Elements
Merge Sort
Quick Sort


Sort Times
20000000
18000000
16000000
14000000
12000000
10000000
8000000
6000000
4000000
2000000
0
Time in Milliseconds
0
20000
40000
60000
80000
100000
120000
Number of Elements
Merge Sort
Quick Sort
Selection Sort