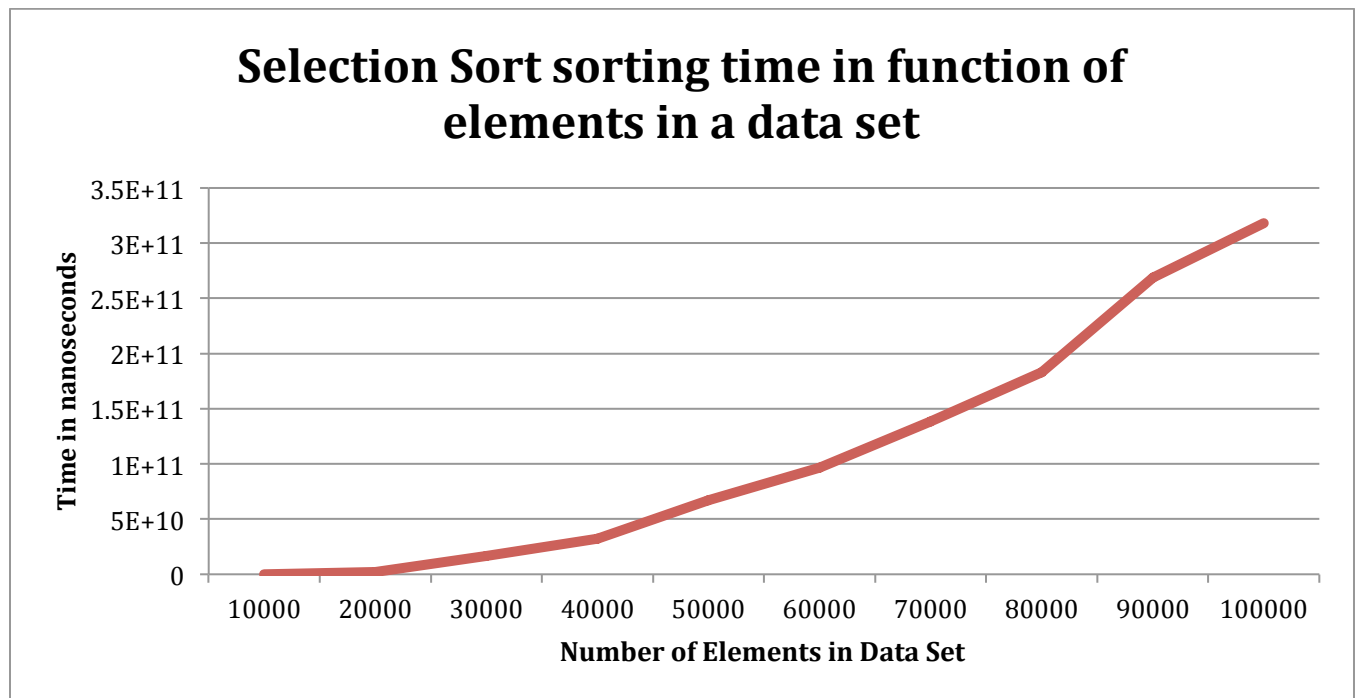


Improving sorting time using different sorting algorithms

We first take a look at the performance of the Selection Sort algorithm when sorting through data sets of different sizes:

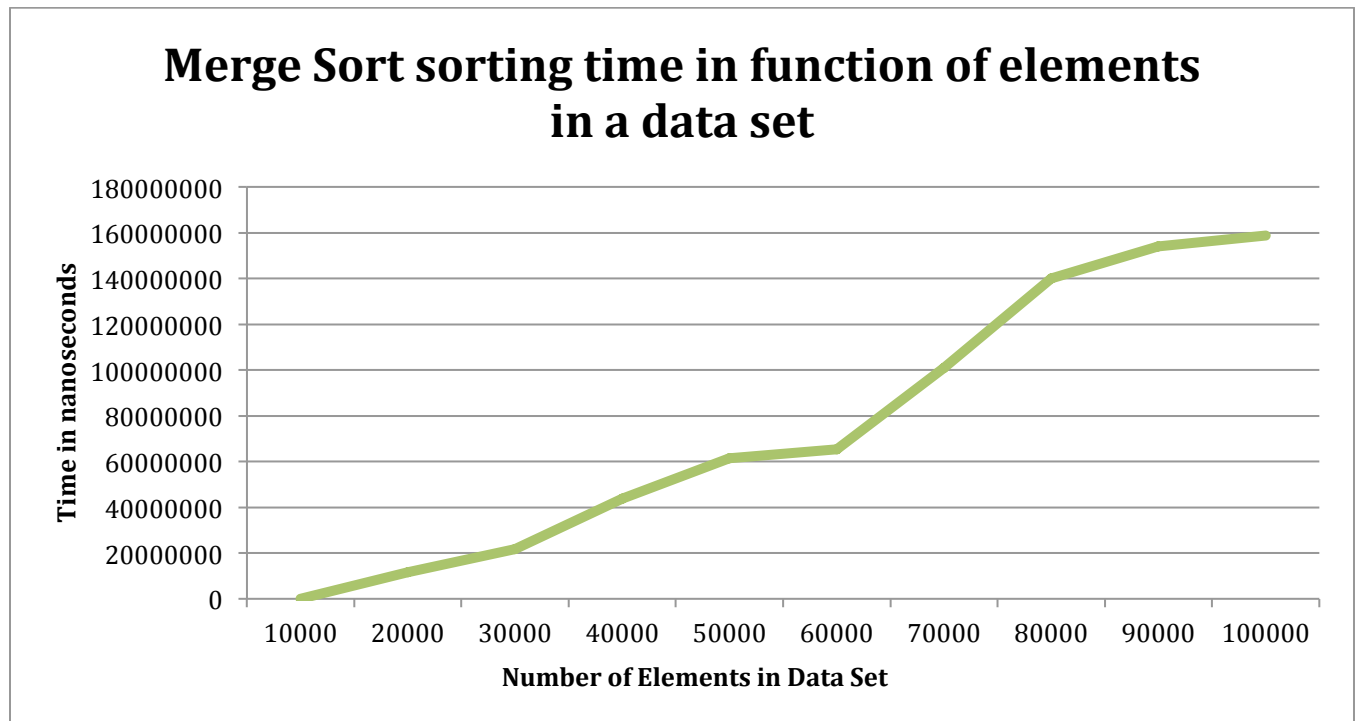
Data Set (Number of elements)	Selection Sort sorting time (in nanoseconds)
10000	23684
20000	2263642380
30000	16871074030
40000	31959570334
50000	67215560030
60000	96747028106
70000	1,38298E+11
80000	1,83071E+11
90000	2,69088E+11
100000	3,18398E+11



As we can see from the graph Selection Sort has an $O(N^2)$ performance. The graph compares the sorting time to the number of element sorted and it shows that the sorting time takes the shape of a quadratic function. We can see an exponential increase in the time it takes for the sort to terminate as the data set gets larger.

We then take a look at the performance of the Merge Sort algorithm when sorting through data sets of different sizes:

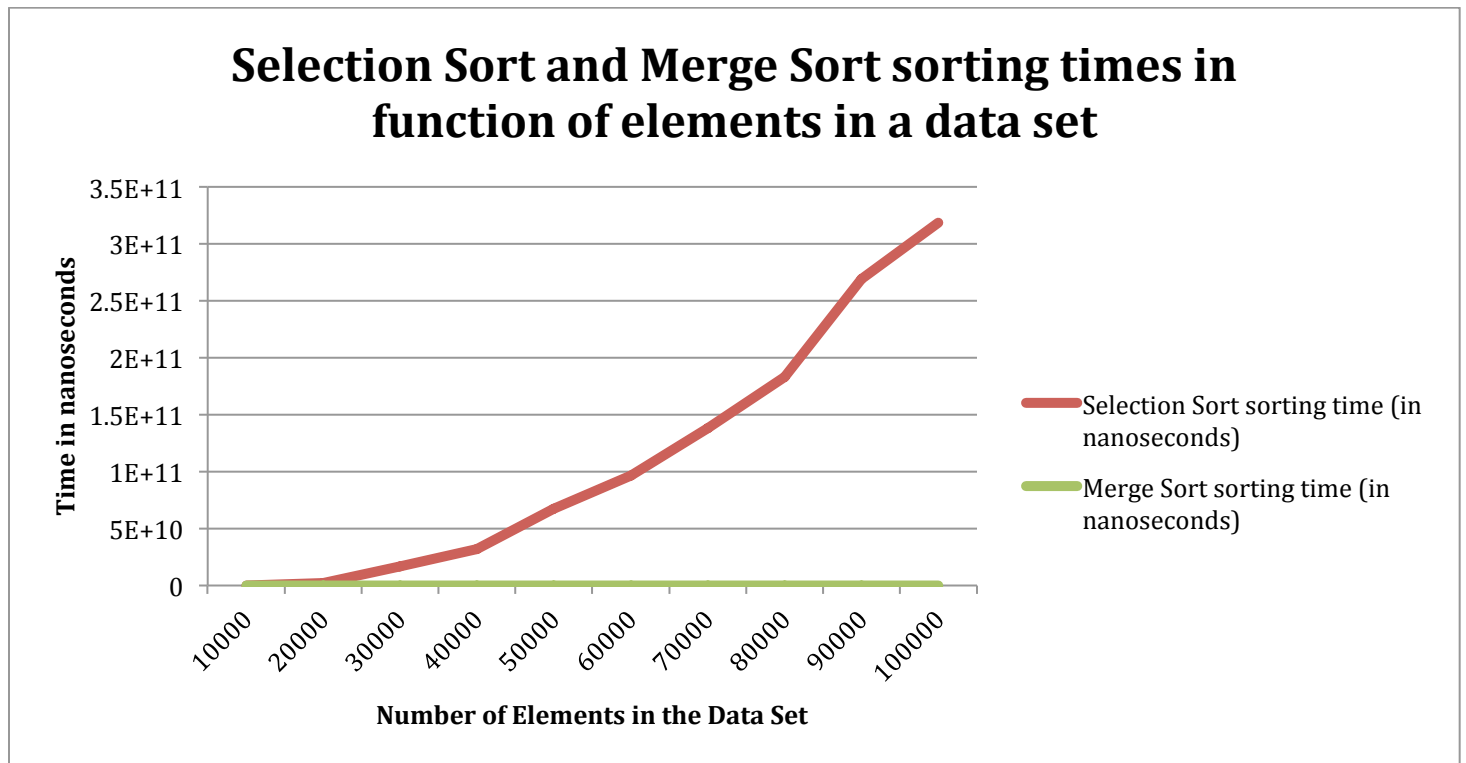
Data Set (Number of elements)	Merge Sort sorting time (in nanoseconds)
10000	19114
20000	11548461
30000	21743994
40000	43731610
50000	61331207
60000	65447637
70000	100795372
80000	140075546
90000	154127193
100000	158715082



As we can see from the graph Merge Sort has an $O(N\log(N))$ performance. The graph compares the sorting time to the number of element sorted and it shows that the sorting time takes the shape of a logarithmic function meaning that the sorting time doesn't increase exponentially as the number of elements in the data set increases (it increases logarithmically).

Finally we compare both sorting times:

Data Set (Number of elements)	Selection Sort sorting time (in nanoseconds)	Merge Sort sorting time (in nanoseconds)
10000	23684	19114
20000	2263642380	11548461
30000	16871074030	21743994
40000	31959570334	43731610
50000	67215560030	61331207
60000	96747028106	65447637
70000	1,38298E+11	100795372
80000	1,83071E+11	140075546
90000	2,69088E+11	154127193
100000	3,18398E+11	158715082



When I first wrote the sort methods I forgot to check for whether the lists were empty before doing the sorting which led into some inconsistencies in sorting times but I was able to fix it by checking for empty lists before sorting them. When we graph the performance of both sorting algorithms it becomes clear that Merge Sort is a much faster sorting algorithm than Selection Sort. This difference becomes especially important when sorting through large data sets. While Merge Sort sorts through a data set of 100 000 elements in about 0.16 seconds it takes Selection Sort 318.4 seconds to do the same operation which is more than a five minute difference. Over small data sets, especially the ones with less than 10 000 elements the difference in sorting times is minimal and not very significant to the overall performance of the program but when Data Sets of 10 000 or more elements need to be sorted Merge Sort performs much better as we expected it to.