WRITEUP
BY:Ruhin Gharai

• What you learned from the different sorting algorithms. Under what conditions do sorts perform well?

Insert sort:- Insertion sort is used when the number of elements is small. It can also be useful when the input array is almost sorted, only a few elements are misplaced in a complete big array.

Heap sort:-The Heap sort algorithm is very efficient. While other sorting algorithms may grow exponentially slower as the number of items to sort increase, the time required to perform Heap sort increases logarithmically.

Quick sort:-Quick sort is an in-place sorting algorithm. In-place sorting means no additional storage space is needed to perform sorting. Merge sort requires a temporary array to merge the sorted arrays and hence it is not in-place giving Quick sort the advantage of space.

Shell sort:- Shell sort is an optimization of insertion sort that allows the exchange of items that are far apart.

Under what conditions do sorts perform poorly?

Insert sort:- when the array is reverse sorted.

Heap sort:- time complexity: n(log)n which is independent of distribution of data.

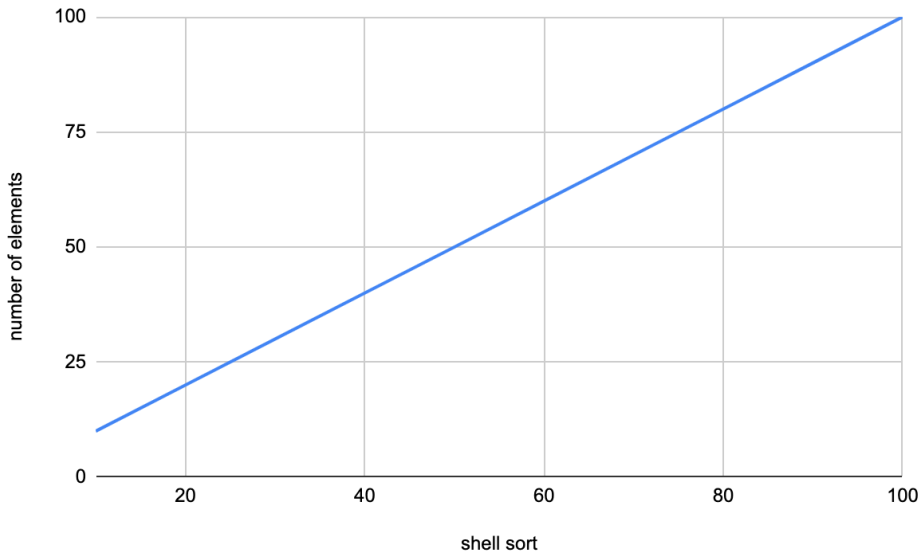
Quick sort:- time complexity of a typical implementation of QuickSort is O(n2). The worst case occurs when the picked pivot is always an extreme (smallest or largest) element. This happens when the input array is sorted or reverse sorted and either the first or last element is picked as pivot.

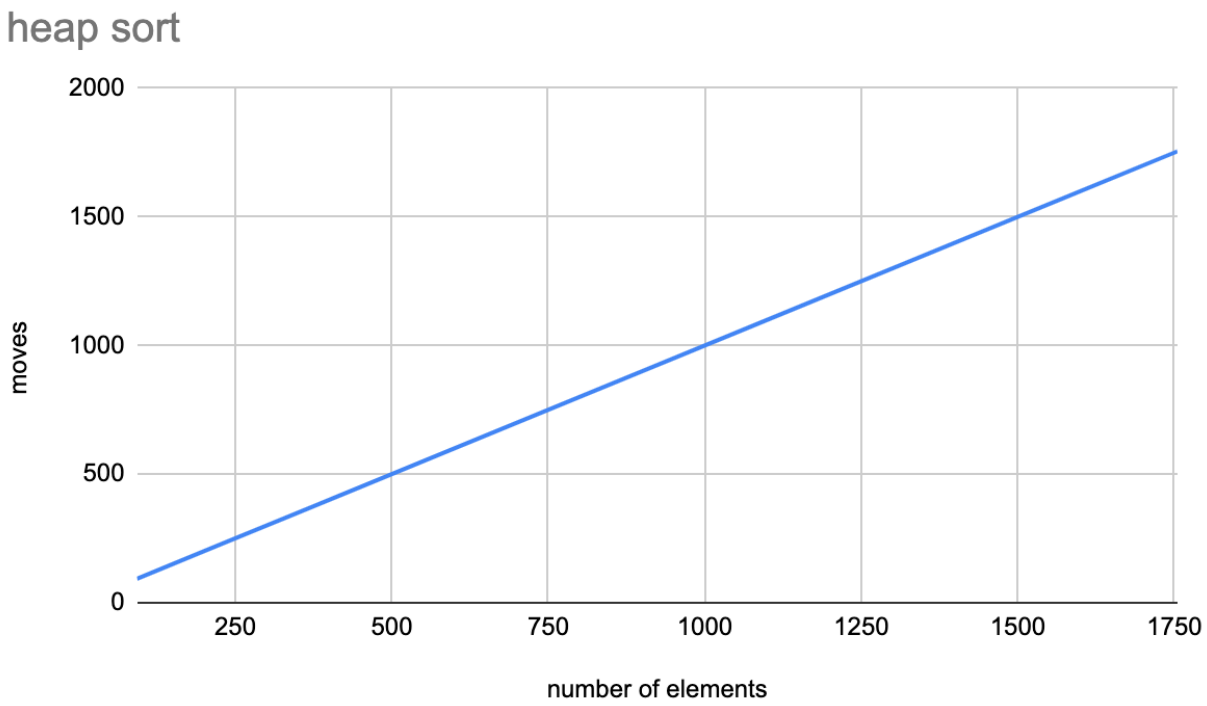
Shell sort:- time complexity worst case is $O(n* \log^2 n)$.

Graphs:-

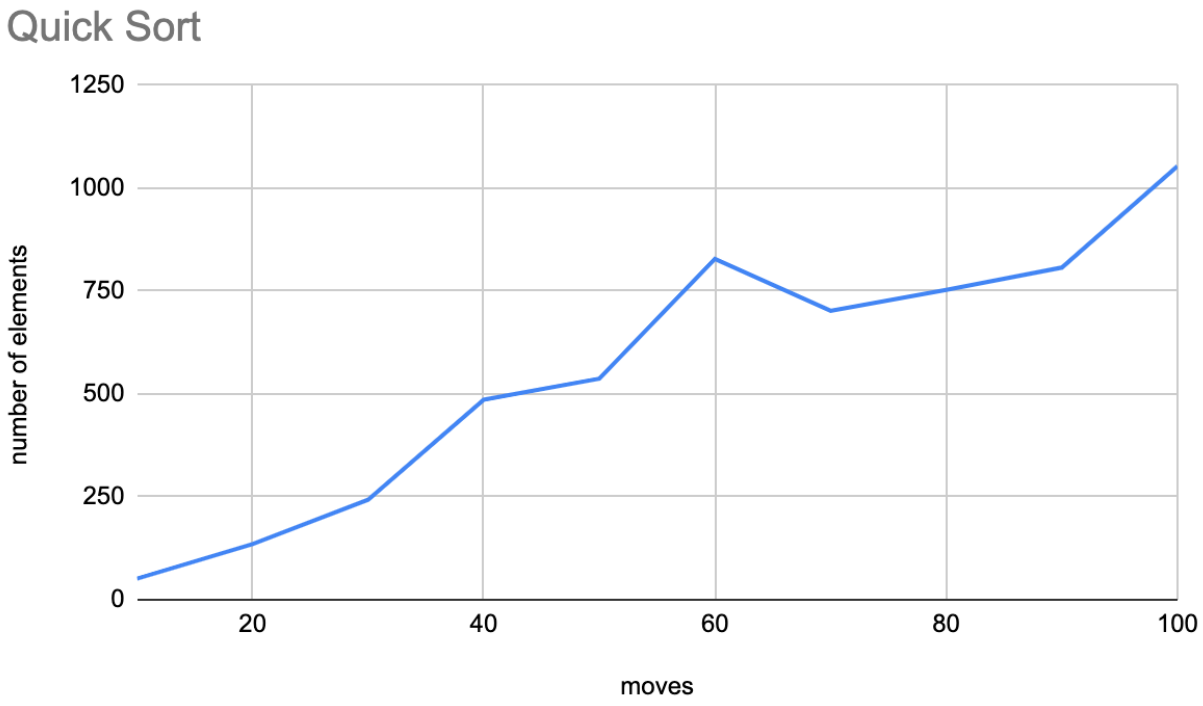
Shell Sort

moves	number of elements
51	10
130	20
223	30
332	40
443	50
603	60
729	70
884	80
1065	90
1183	100

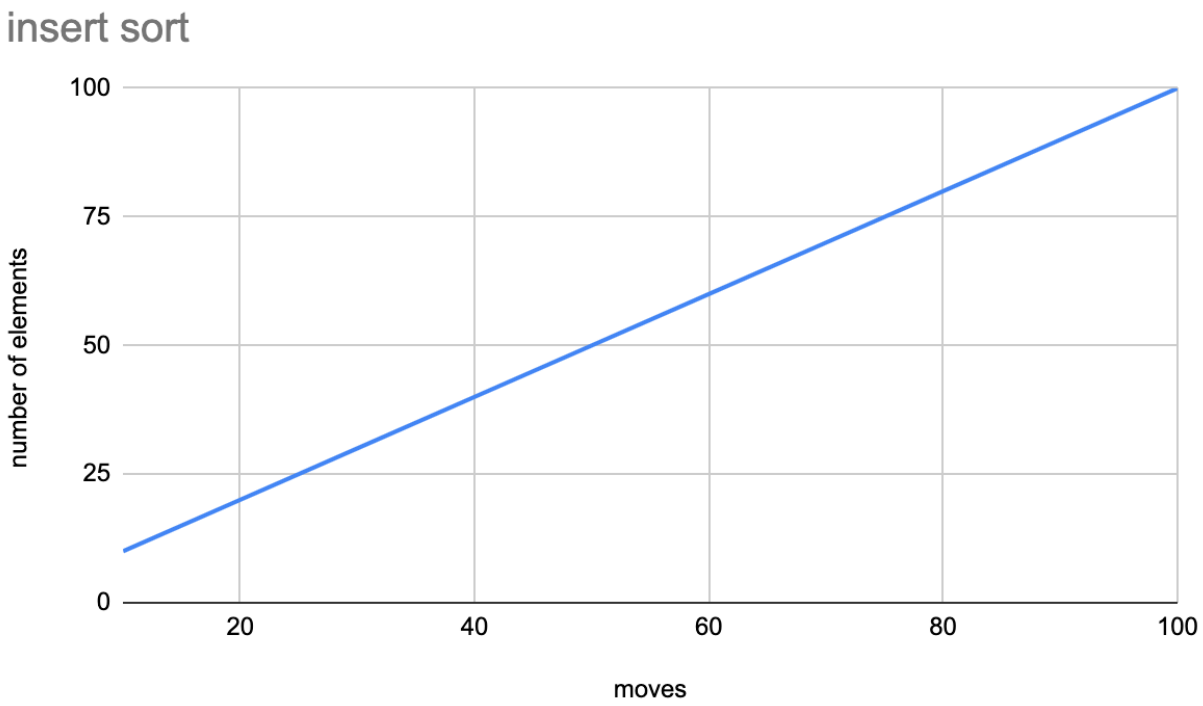




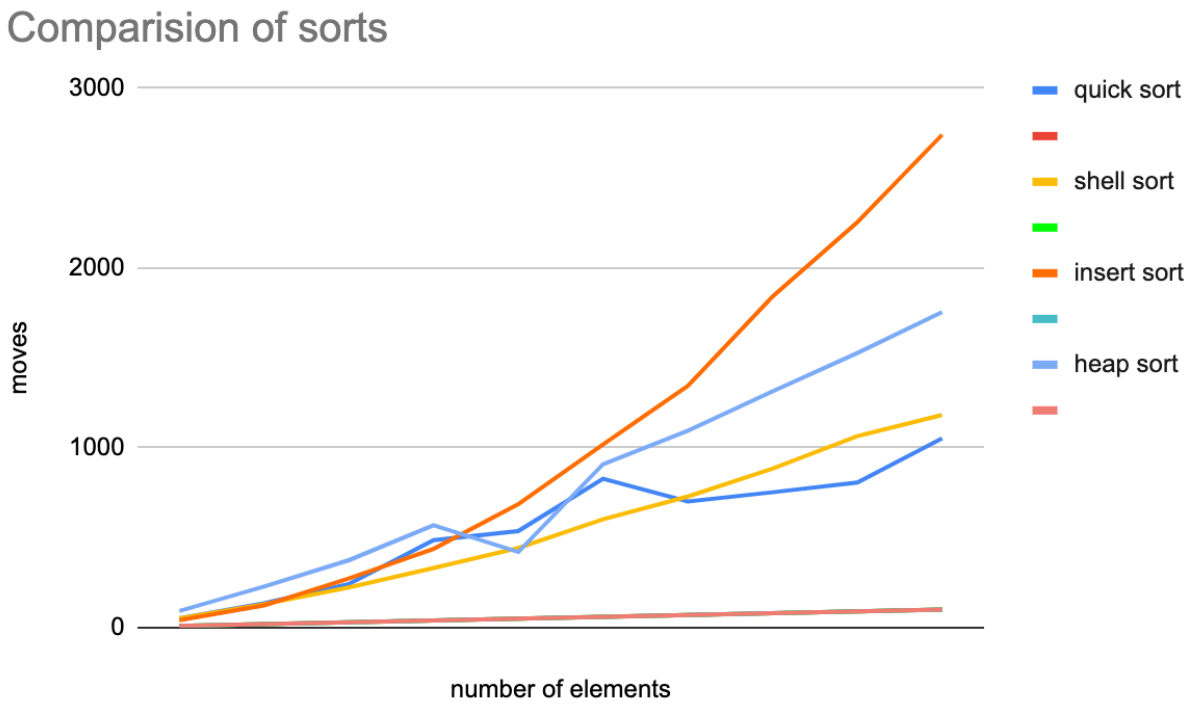
moves	number of elements
93	10
228	20
375	30
570	40
421	50
909	60
1095	70
1314	80
1527	90
1755	100



moves	number of elements
51	10
135	20
243	30
486	40
537	50
828	60
702	70
753	80
807	90
1053	100



moves	number of elements
41	10
124	20
273	30
438	40
687	50
1019	60
1345	70
1842	80
2255	90
2741	100



What conclusions can you make from your findings?

As one can see from the graph comparison and the values given above for the individual graph for each sort along with the values for the given graphs of the moves and the number of elements to arrange and sort each element in order we can see insert sort and shell sort are more accurate compared to quick and heap sort. Shell sort is more accurate in ordering compared to insert cort as there are less differences between the numbers(showing less bumps in the lines in the graph).So with proof of the graphs given above yes they do all sort , but the accuracy is best with shell sort making it the better sorting algorithm.

The efficiency is really good for all sorts as each provided a straight line except quick sort as it was crooked with the individual graph and merged graph showing that it is quick to use but not the best sorting algorithm.

I learned a lot of each type of sorting algorithm for this class which one is better compared to the other, which will definitely help in future programs