Report for lab number 1 Data Structure

Topic:

Difference between various sorting algorithms regarding time of execution

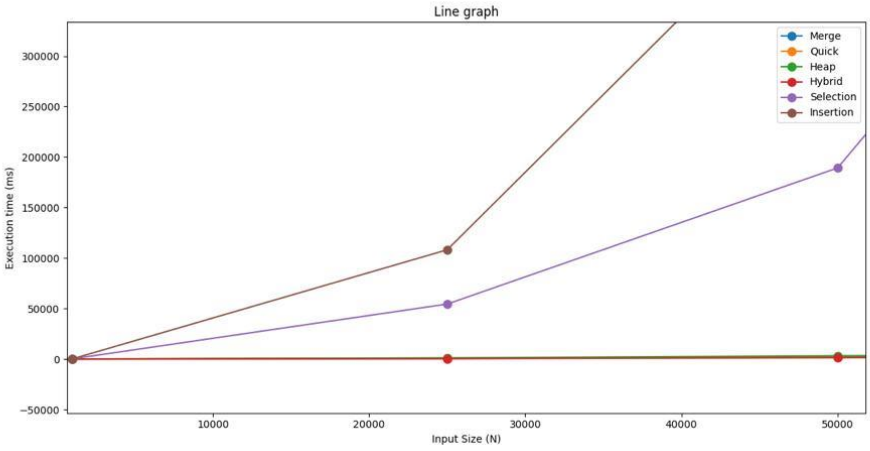
By students:

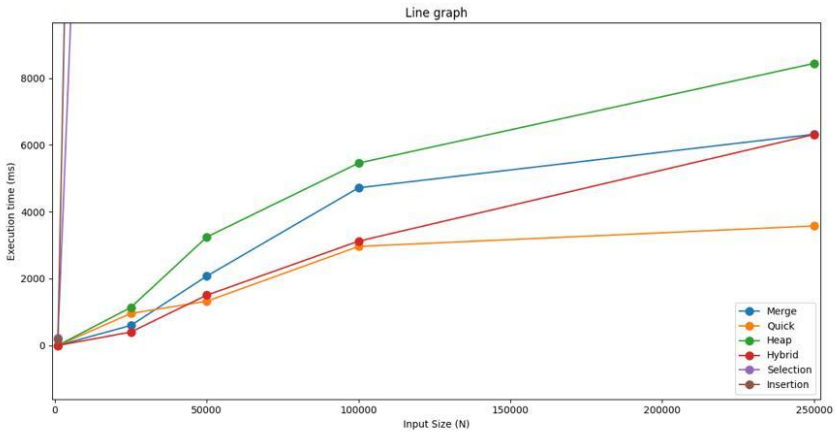
Yassin Medhat Helmi 7574
Hana Waleed Farouk 7599
Fady Sarwat Farouk 7432

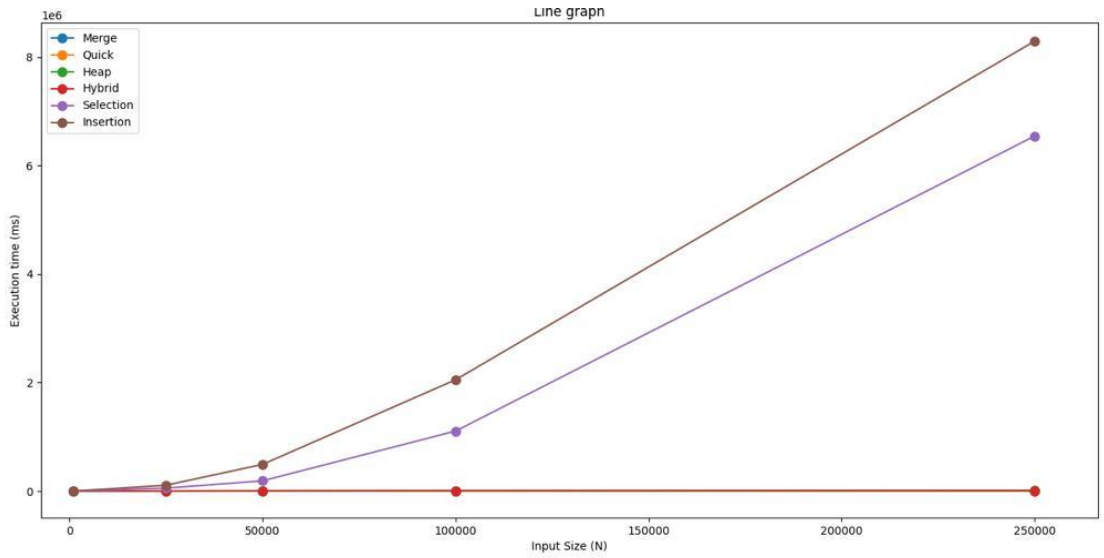
In this lab assignment we were asked to implement the following sorting algorithms and compare their execution time:

Merge sort algorithm O(nLogn)
Quick Sort algorithm O(nLogn)
Hybrid sort (mix between selection and merge sort) algorithm O(nLogn)
Heap sort algorithm O(nLogn)
Selection sort algorithm O(n^2)
Insertion sort algorithm O(n^2)

The time vs size of array plots:







Data of plot written:

Algorithm/ Size	1000 elements	25,000 elements	50,000 elements	100,000 element	250,000 element
Insertion	150.9611 msec	110699.5 msec	414248.583 msec	1736509.9 msec	8826843.53 msec
Selection	66.00594 msec	51184.24 msec	184964.542 msec	794351.9 msec	6500574.18 msec
Merge	3.02814 msec	660.0003 msec	1500.552 msec	3108.948 msec	8725.730 msec
Quick	2.00104 msec	470.0036 msec	1076.557 msec	3136.818 msec	7515.399 msec
Heap	3.97086 msec	1026.386 msec	3049.51882 msec	6279.012 msec	10391.18 msec
Hybrid	2.00390 msec	359.0977 msec	1631.161 msec	2838.948 msec	8570.226 msec

As shown in the table below the worst-case algorithm is the Insertion sort
Also you can notice how Merge, Quick, Heap, Hybrid sort Algorithms are all with complexity O(nLogn) and still score differently from each other
that is because of the difference in their best cases and the conditions of the array e.g., repetition in numbers.

In addition, the quick sort is shown to be working efficiently BUT we shouldn't forget that it is unstable and can break down in some cases with complexity of O(n^2)

IN CONCLUSION:
All sorting algorithms are efficient in their own way and has benefits in their best environment, the key is to know which one to use regarding the conditions of the data you are dealing with.