Lab 1 data structure

Topic:time complexity of various sorting algorithm

Grp 4 section 1

student name-ID:

peter anton-7406

zeyad ahmed-7621

Abdelrahman ahmed reefat-7433

We have implement the following algorithm so we will clarify it and it worst case time complexity:

1-heap sort→O(nlogn)

2-merge sort→O(nlogn)

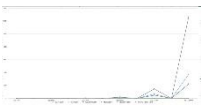
3-quick-sort →O(nlogn)

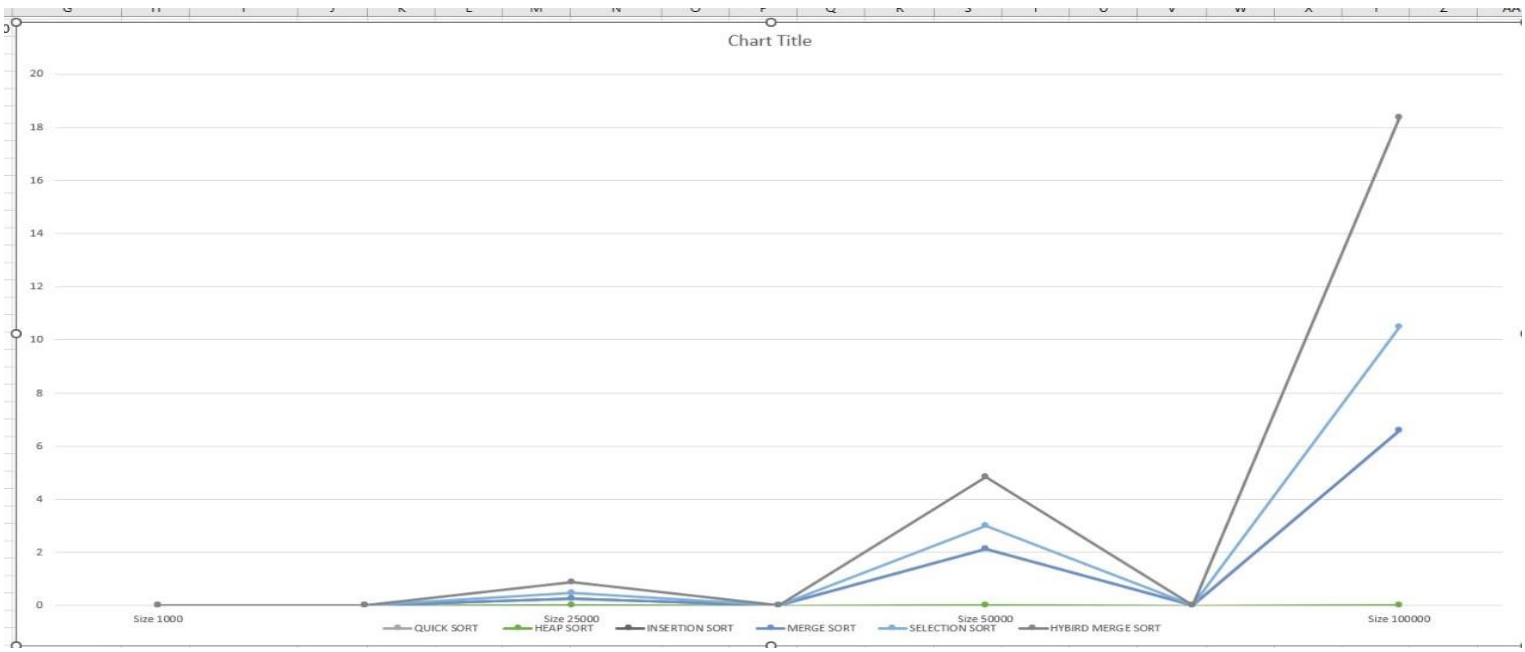
3-insertion sort→$O(n^2)$

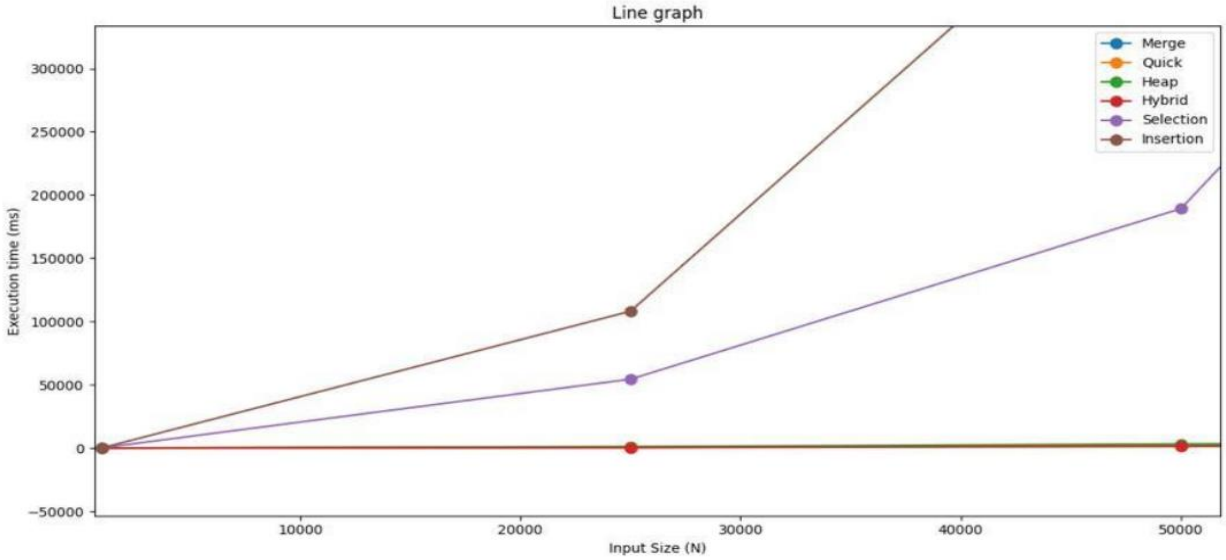
4-hybrid sort(merge sort then selection sort)→O(nlogn)

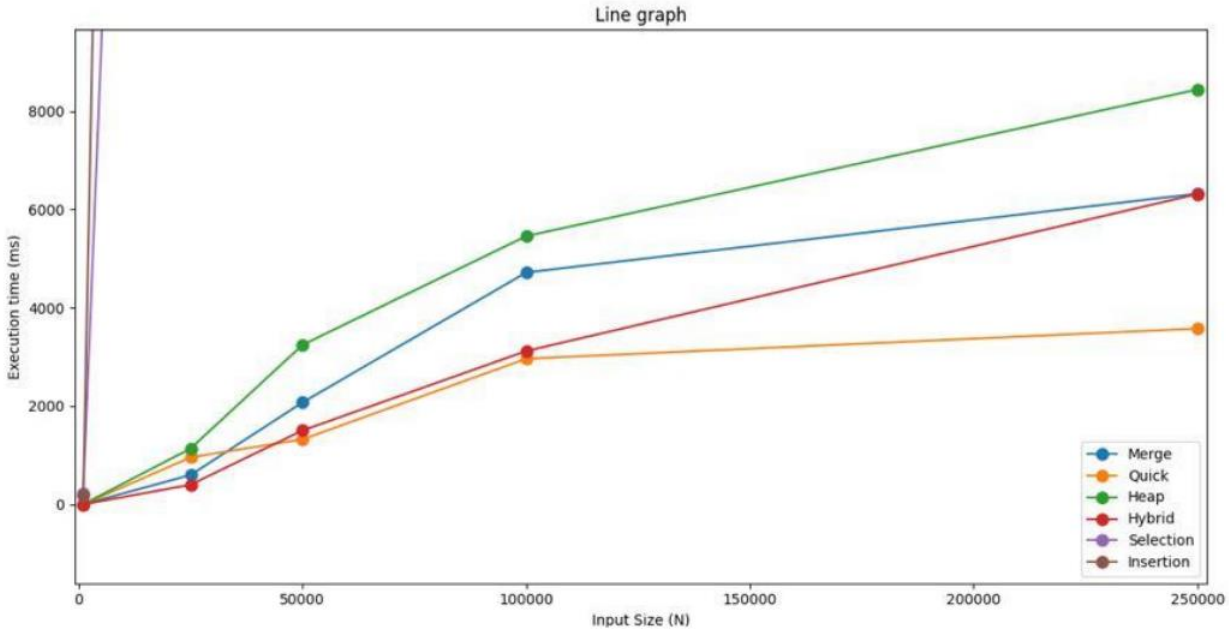
5-selection sort→$O(n^2)$

Sort/array Size	1 k	25K	50K	100K	250K
Heap sort	2999.30 µs	0.001990 473 min	0.00476 166 min	0.01276 844 min	0.02729 607 min
Merge sort	1537.79 98 µs	0.001100 973 min	0.00342 32min	0.00543 737 min	0.01292 334 min
Selection sort	0.00030 580 min	0.206332 958 min	0.87488 481 min	3.88382 101min	20.4343 759 min

Insertion sort	0.0003820 min	0.265140037 min	2.11231362 min	6.5591886 min	46.346740 min
Hybrid sort	0.0006316 min	0.416653061 min	1.83310277min	7.9009518 min	79.2041628 min
Quick sort	2002.95448 µs	0.000990673 min	0.00216920 min	0.00533867 min	0.0107279  min







as we see in the previous graph and table the insertion and selection sort are so bad in worst time complexity

also we notice that merge , heap and quick sort are all O(n log n) but they differ in time (according to the table)

that’s because of the difference in their best cases and the condition of the array

also the quick sort look like it will be the fastest one and work efficiently but we forget that it is unstable and can break down some cases with complexity of $O(n^2)$

So finally the sorting algorithm are efficient in their own way and we should know where and when we use it