Names: Youssef Mohamed Elraggal
Mohamed Osama Fathi

Group 2 Section 1

Assignment 1 Data Structure

-Brief

- The project aim is to test and plot different sorting algorithms and compare between them in a good, visualized graph. Also Merging between merge sort and selection sort. Finally, Finding Kth smallest element using partitioning method.
- The Project is modified with GUI Interface for easier user usage.
- The code is written in Python

-Graph for Sorting Algorithms Performance

-Entering Samples of [10, 5000, 15000, 25000, 50000] in Fig1.1

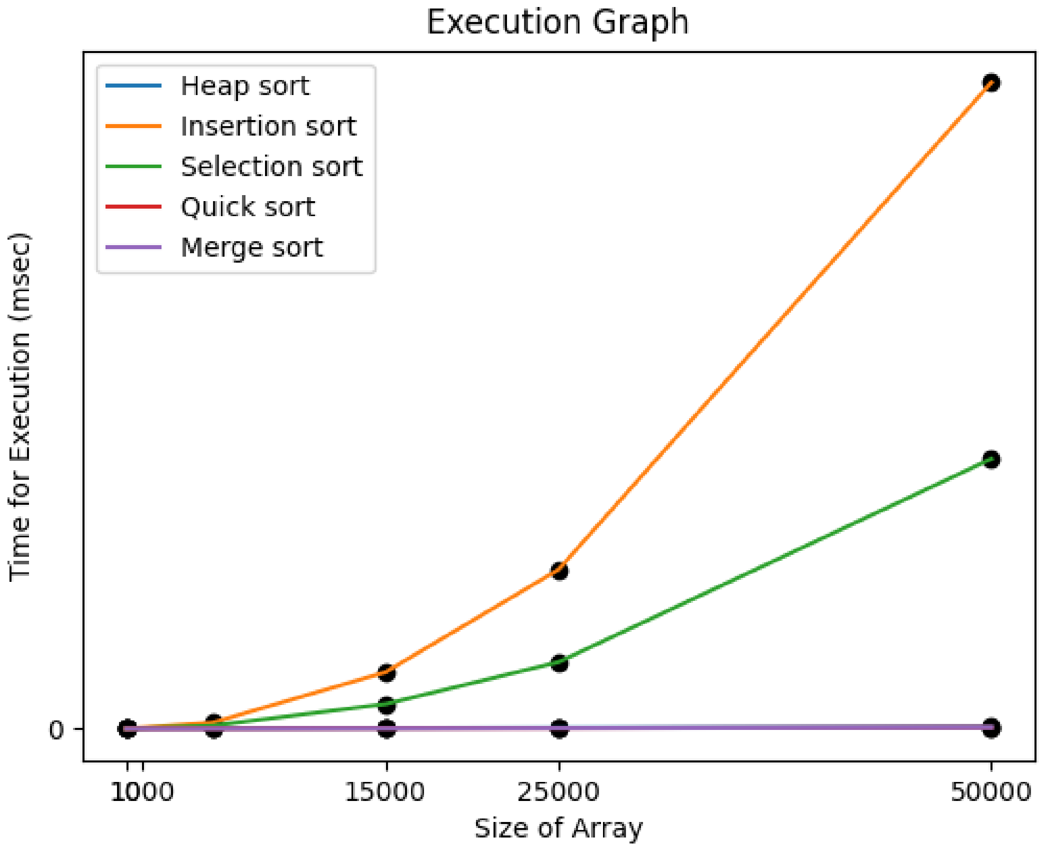
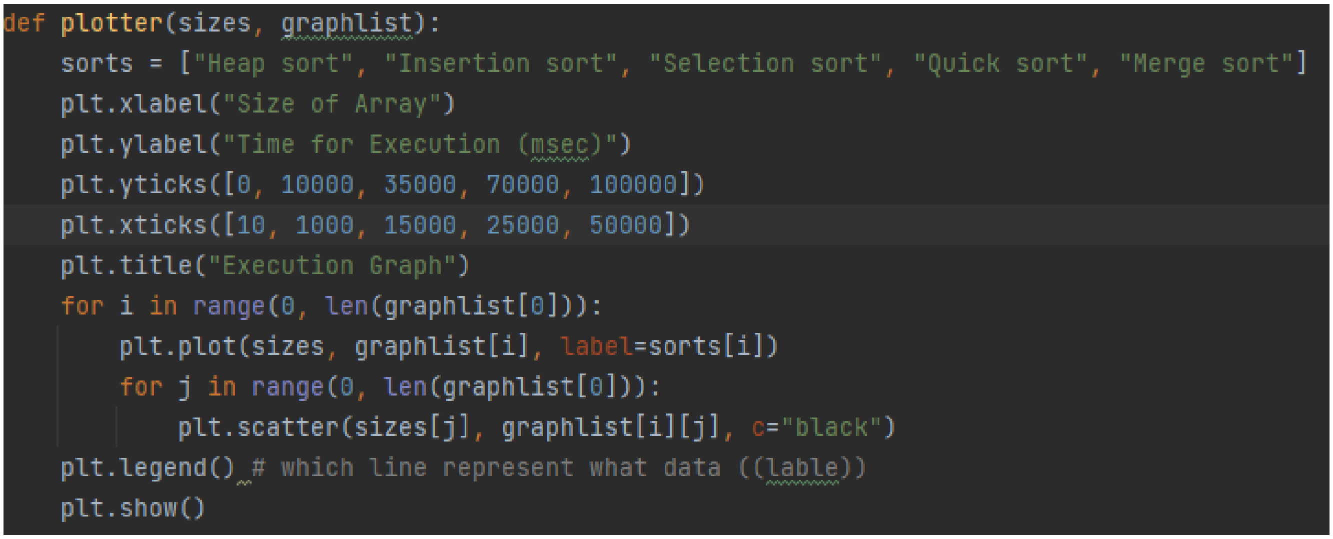


Fig 1.1

-The graph is produced using Matplotlib package in python is shown above in function “Plotter” in Fig 1.2



```
def plotter(sizes, graphlist):
    sorts = ["Heap sort", "Insertion sort", "Selection sort", "Quick sort", "Merge sort"]
    plt.xlabel("Size of Array")
    plt.ylabel("Time for Execution (msec)")
    plt.yticks([0, 10000, 35000, 70000, 100000])
    plt.xticks([10, 1000, 15000, 25000, 50000])
    plt.title("Execution Graph")
    for i in range(0, len(graphlist[0])):
        plt.plot(sizes, graphlist[i], label=sorts[i])
        for j in range(0, len(graphlist[0])):
            plt.scatter(sizes[j], graphlist[i][j], c="black")
    plt.legend() # which line represent what data ((lable))
    plt.show()
```

Fig 1.2

-Specific Graph to show difference between Merge , Quick Sort ,and Heap Sort in Fig1.3

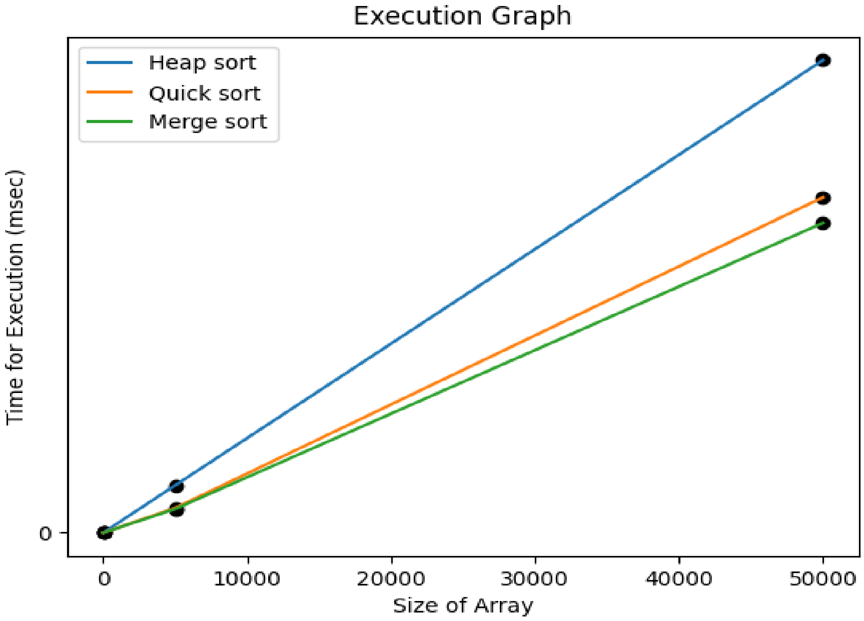


Fig1.3

Here Shown Table With no of samples inserted and time taken by every algorithm to sort them.

Time in Ms

No.Samples	Heap	Insertion	Selection	Quick	Merge
10	0.0	0.0	0.0	0.0	0.0
5000	35.9	1372.15	735.5	24.9	25.9
15000	125.9	14412.1	6018.2	88.9	79.95
25000	223.8	38561.2	17181.4	157.9	143.9
50000	473.3	151120.6	69404.7	327.7	302.8