Lab Report-1
DAA

NAME-PRIYANSH BAKHTANI

PRN-20190802079

LAB-1

Aim: To study and compare time complexity of insertion sort,bubble sort,selection sort,quick sort and merge sort and to make their graphs using experimental values.

1.)**Selection sort**:The selection sort algorithm sorts an array by repeatedly finding the minimum element (considering ascending order) from unsorted part and putting it at the beginning.

2.)**Bubble sort:**Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in wrong order.

3.)**insertion sort:**Insertion sort is a simple sorting algorithm that works similar to the way you sort playing cards in your hands. The array is virtually split into a sorted and an unsorted part. Values from the unsorted part are picked and placed at the correct position in the sorted part.

4.)**quick sort:**QuickSort is a Divide and Conquer algorithm. It picks an element as pivot and partitions the given array around the picked pivot.

5.)**merge sort:** merge sort divides input array in two halves, calls itself for the two halves and then merges the two sorted halves.

In this experiment we took a randon list of size n=10,100,1000,10000 and used these arrays for different sorting algorithms to calculate its run time.

Observations:

1.)Bubble sort is the slowest algorithm.

2.)Quick sort is more efficient incase of smaller arrays where as merge sort is more efficient in case of larger arrays.

3.)The other two arrays(insertion sort and selection sort)takes average time to sort.

Graphs:

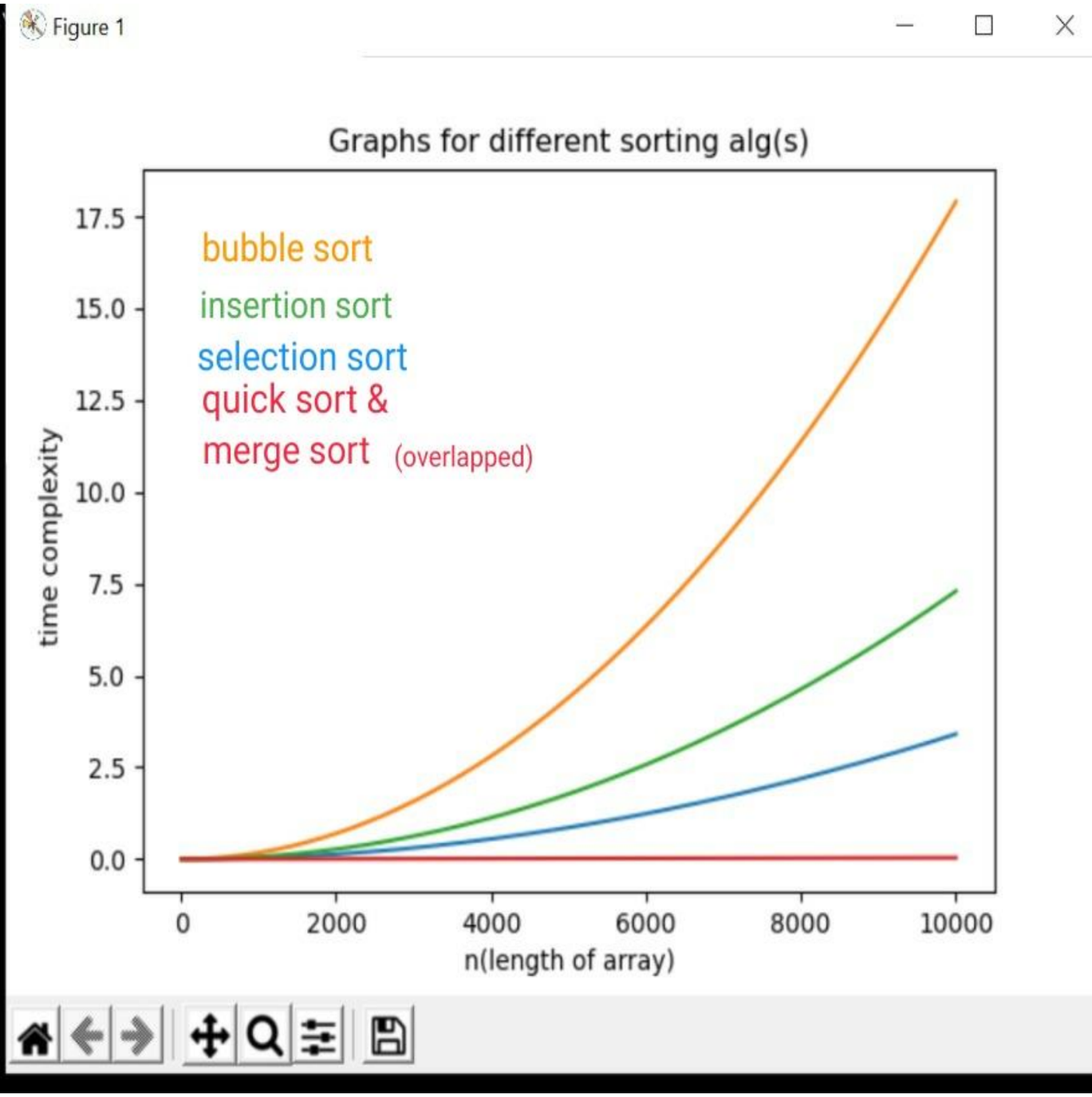
```
C:\Users\HP\priyansh_bakhtani\venv\Scripts
selection sort time complexities
for n=10 r1= 0.0
for n=100 r2= 0.0
for n=1000 r3= 0.031912803649902344
for n=10000 r4= 3.406101703643799

Bubble sort time complexities
for n=10 b1= 0.0
for n=100 b2= 0.0019943714141845703
for n=1000 b3= 0.17553114891052246
for n=10000 b4= 17.905998945236206

insertion sort time complexities
for n=10 i1= 0.0
for n=100 i2= 0.0
for n=1000 i3= 0.06382918357849121
for n=10000 i4= 7.3003990650177

quick sort time complexities
for n=10 q1= 0.0
for n=100 q2= 0.0
for n=1000 q3= 0.0029954910278320312
for n=10000 q4= 0.03988790512084961

merge sort time complexities
for n=10 m1= 0.0
for n=100 m2= 0.0
for n=1000 m3= 0.00598454475402832
for n=10000 m4= 0.0727987289428711
```



Conclusion: Hence we calculated the practical runtime values and obtained a parabolic curve(n^2) in case of selection sort, bubble sort and insertion sort and and straight line curve (nlogn) in case of quick sort and merge sort which exactly same as the theoretical values and graph.