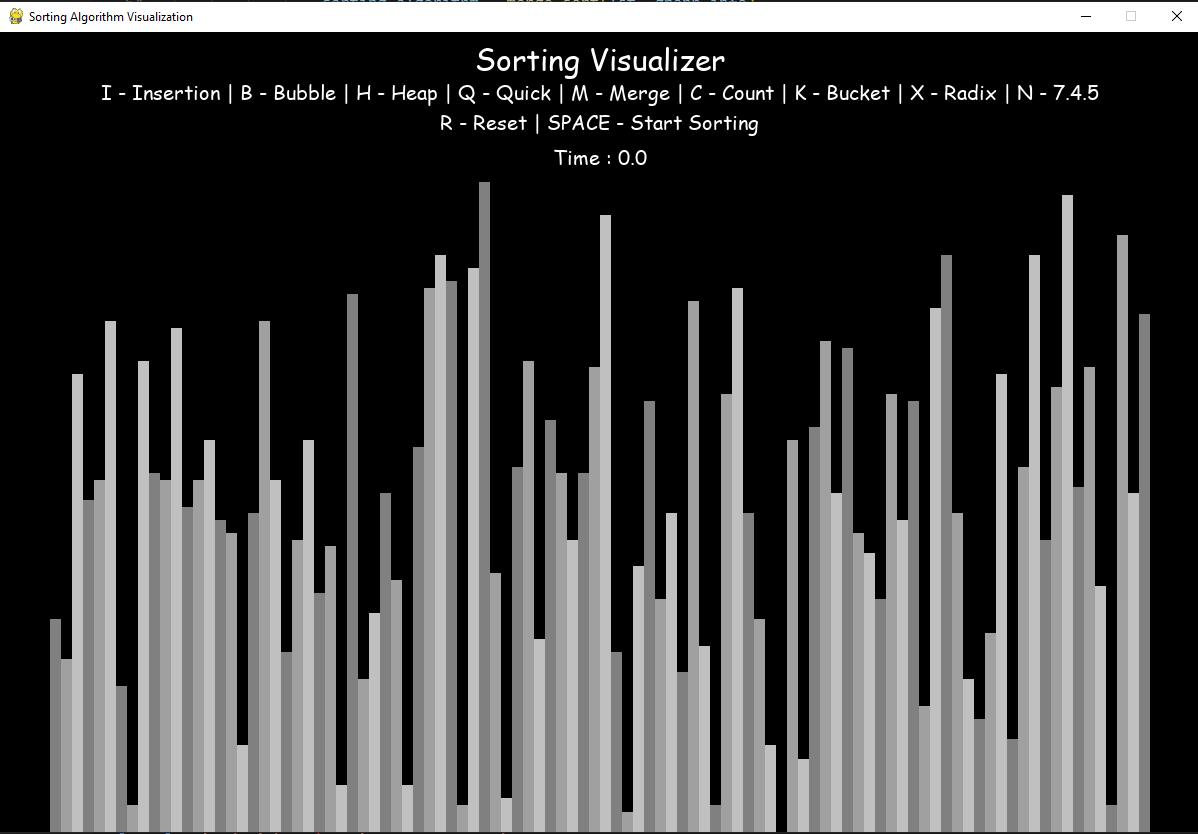
ABSTARCT:

This project is about a sorting visualizer, which would greatly help new students or anyone who is new to sorting techniques better understand the concepts.

Introduction and Language:

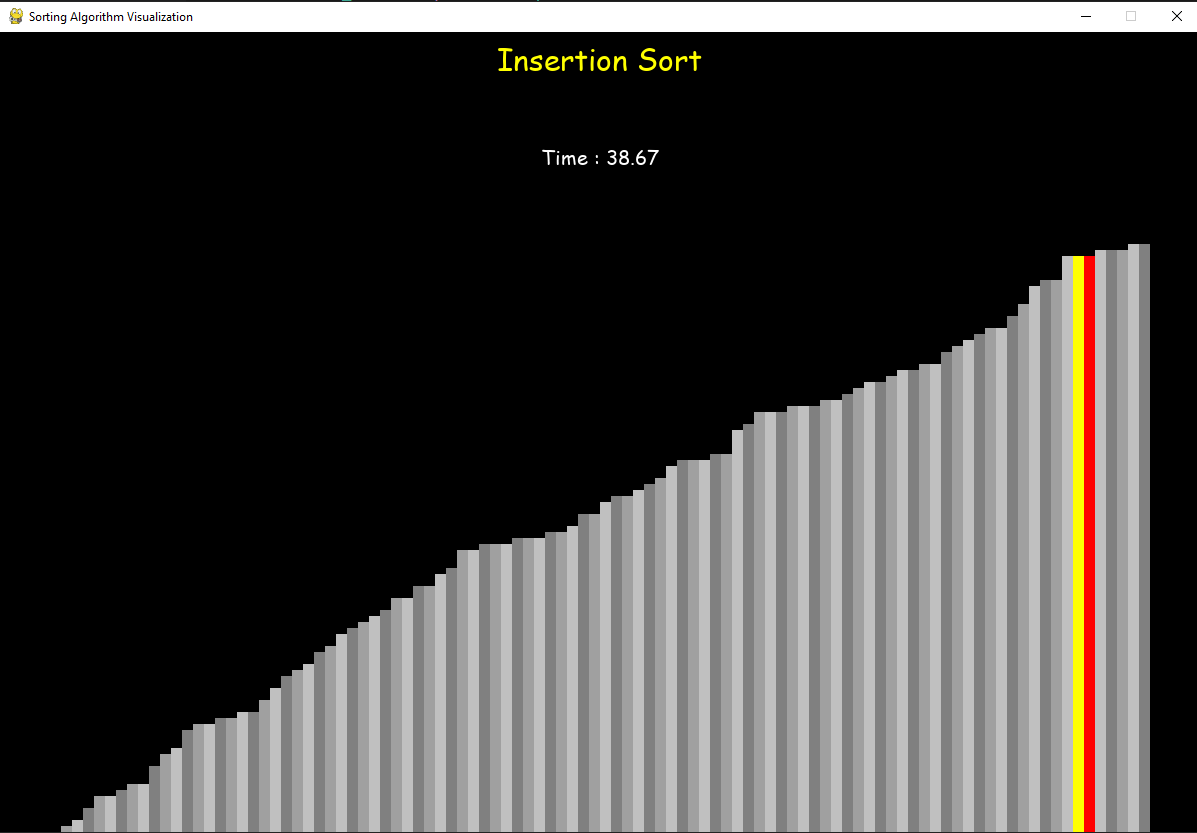
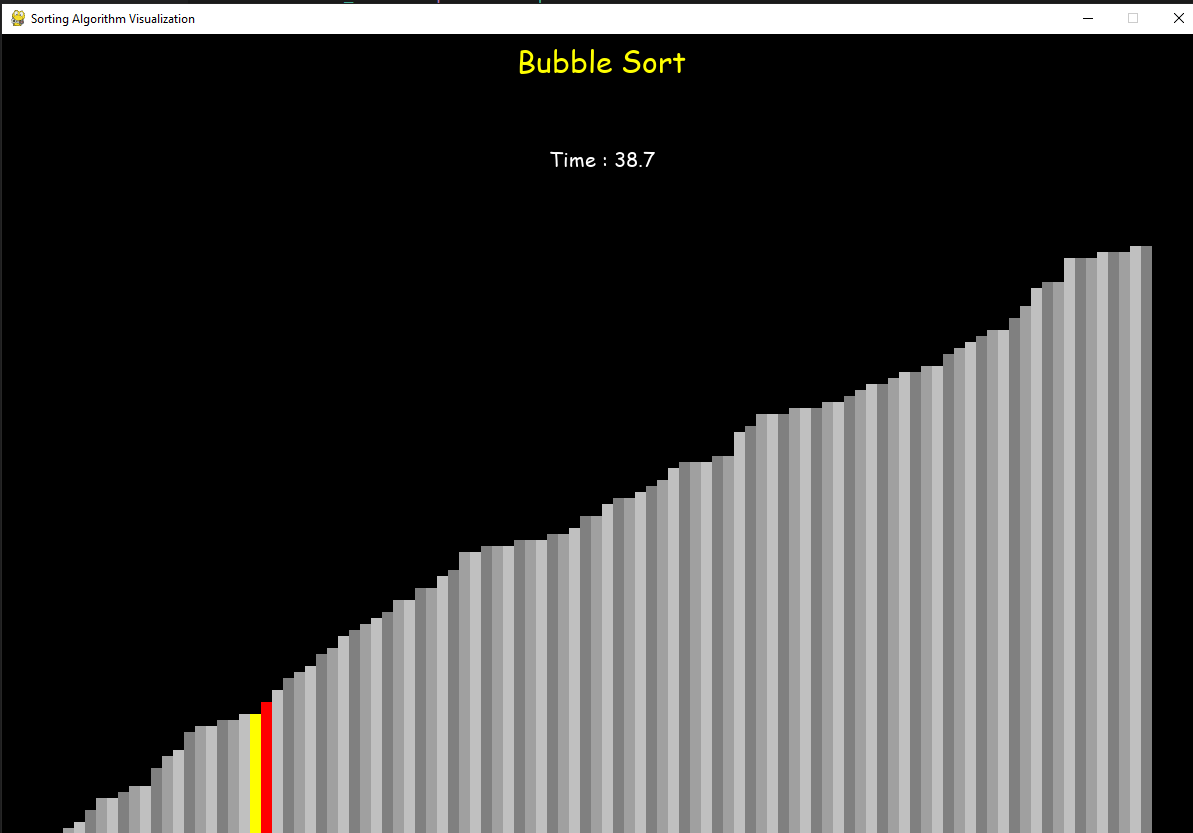
This is a sorting algorithm visualizer which is built using python3 and pygames. Random numbers are taken from a file and which are in turns used to draw the bar graph. The program run at the initial with the bubble sort, we can change any sort we want to implement using the controls option given above the program. A bar graph is shown for the numbers generated and which shifts according to the algorithm used to sort the given list.

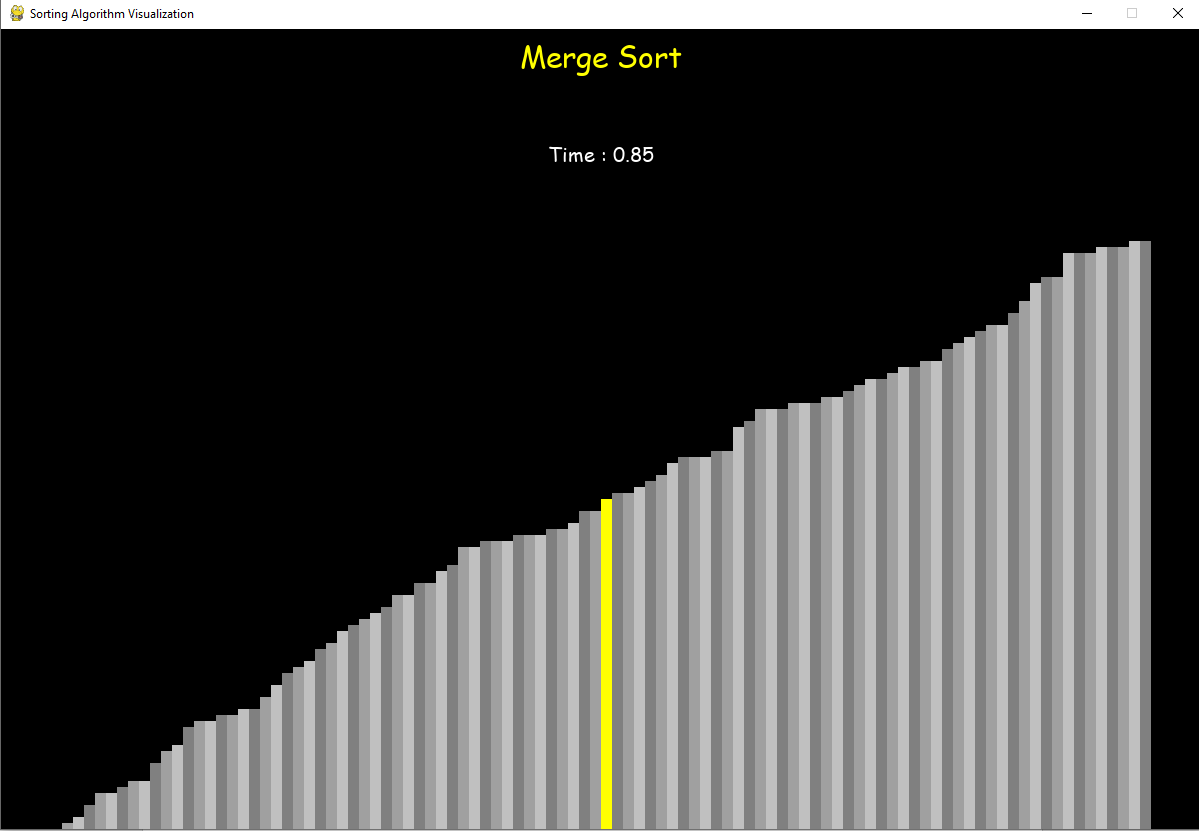
Experimental Setup:

We are using visual studio code which is helping us in compiling and running the python code on a console.

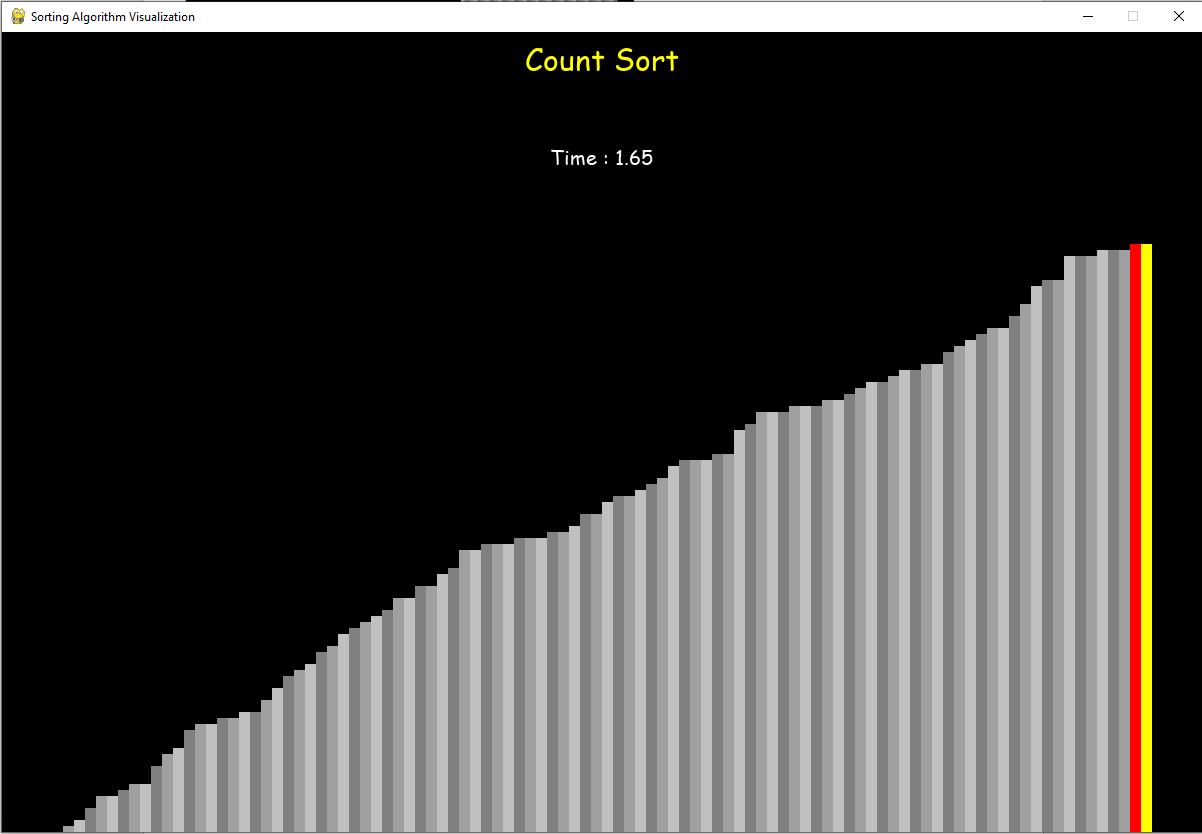
Result:

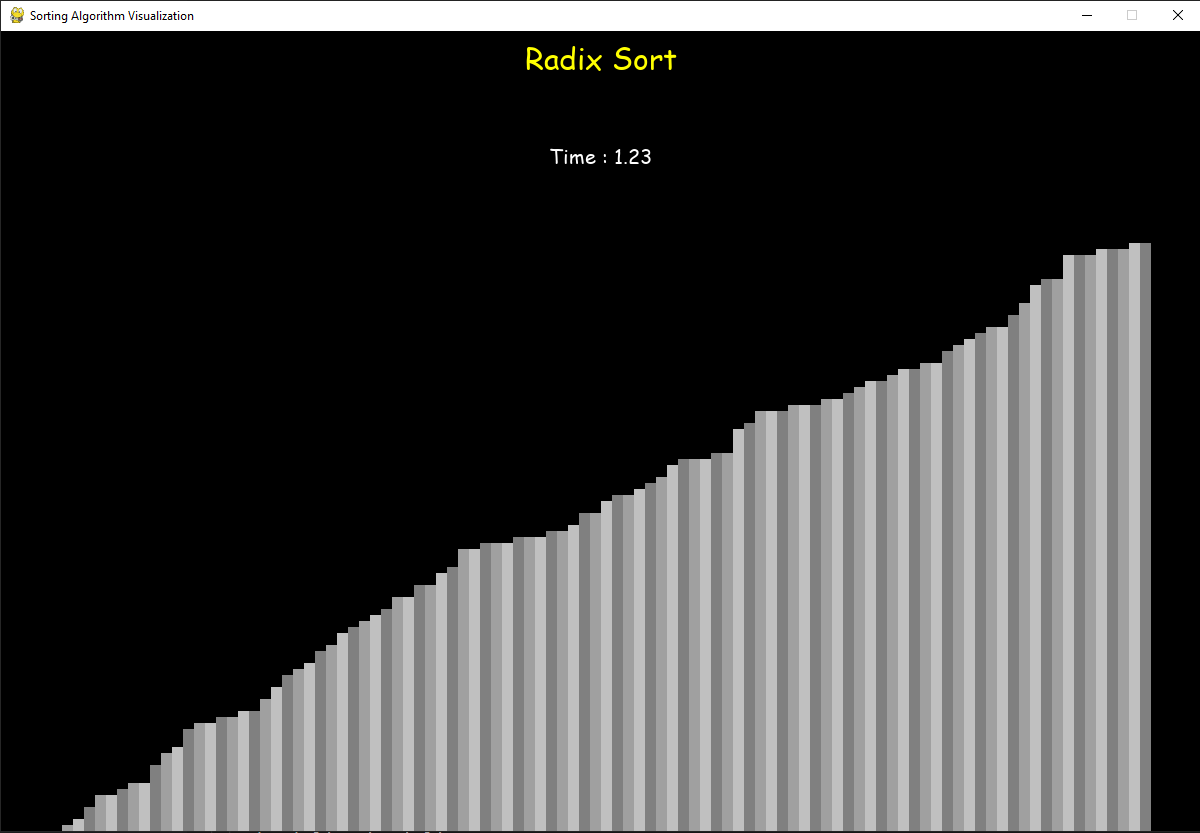
All the Sorting algorithms, for testing purposes, are ran on the smallest file of 100 numbers. We have 3 files of numbers number\_s.txt which consist 100 numbers, number\_m.txt which consist 250 numbers and number\_l.txt which consist 500 numbers. Here in result after running the time displayed for all the algorithm are shown.

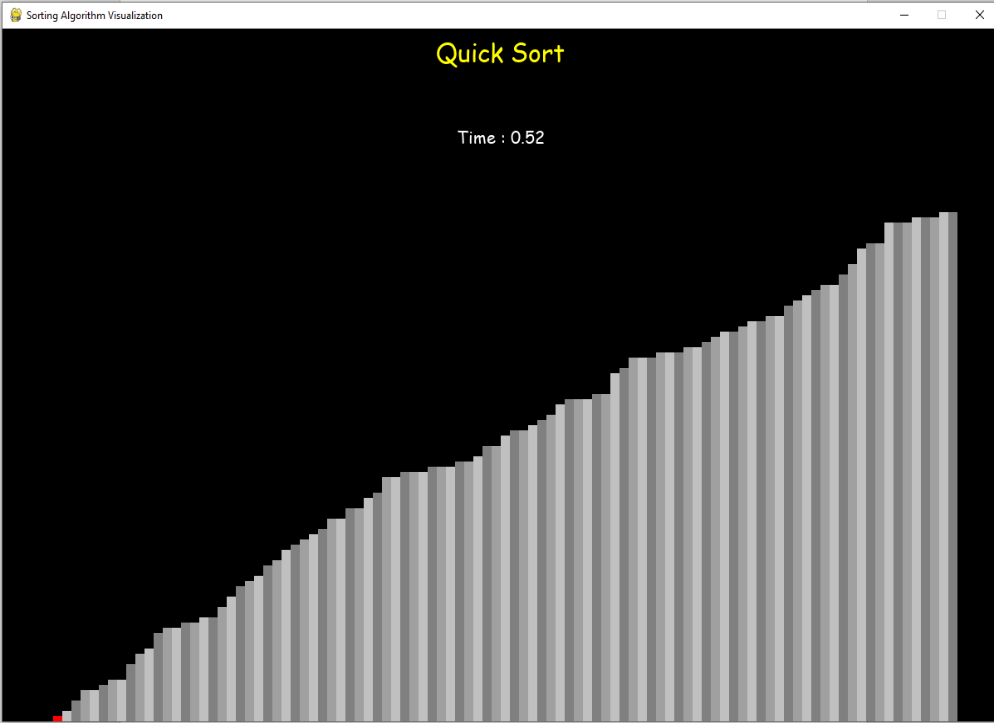


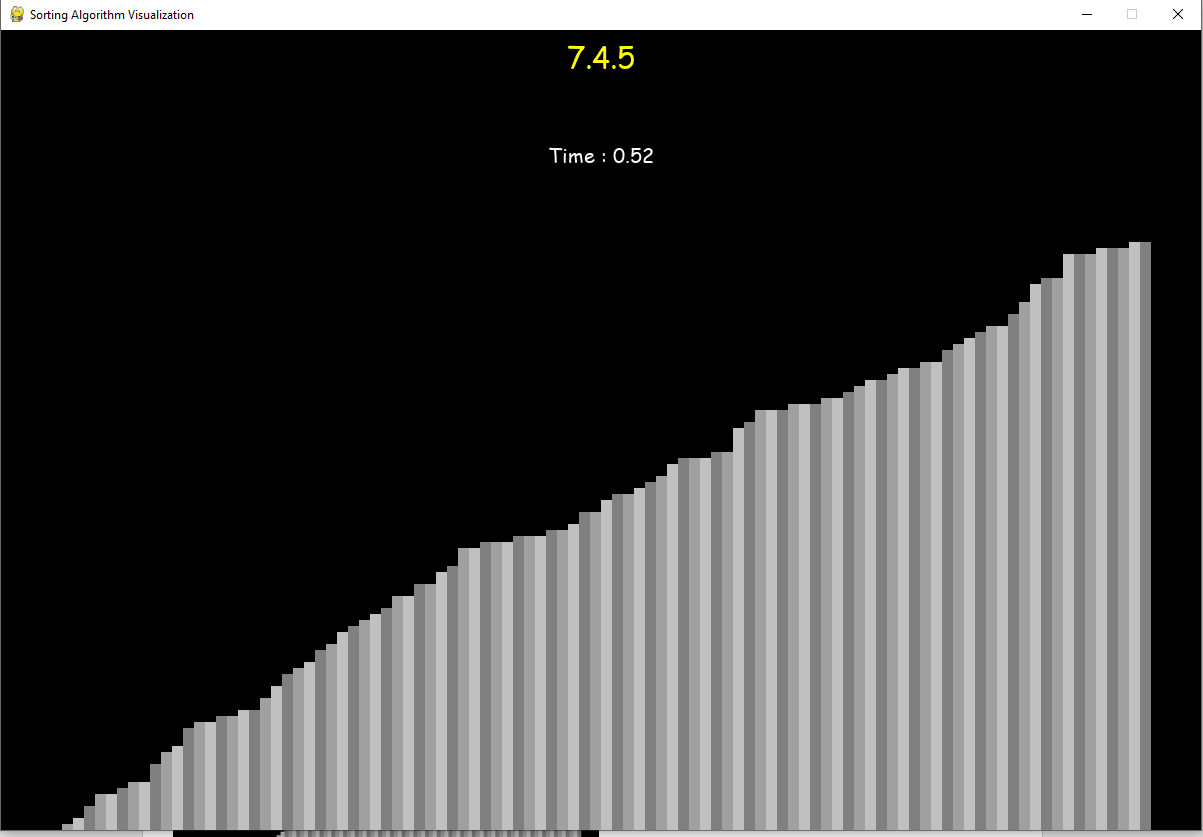


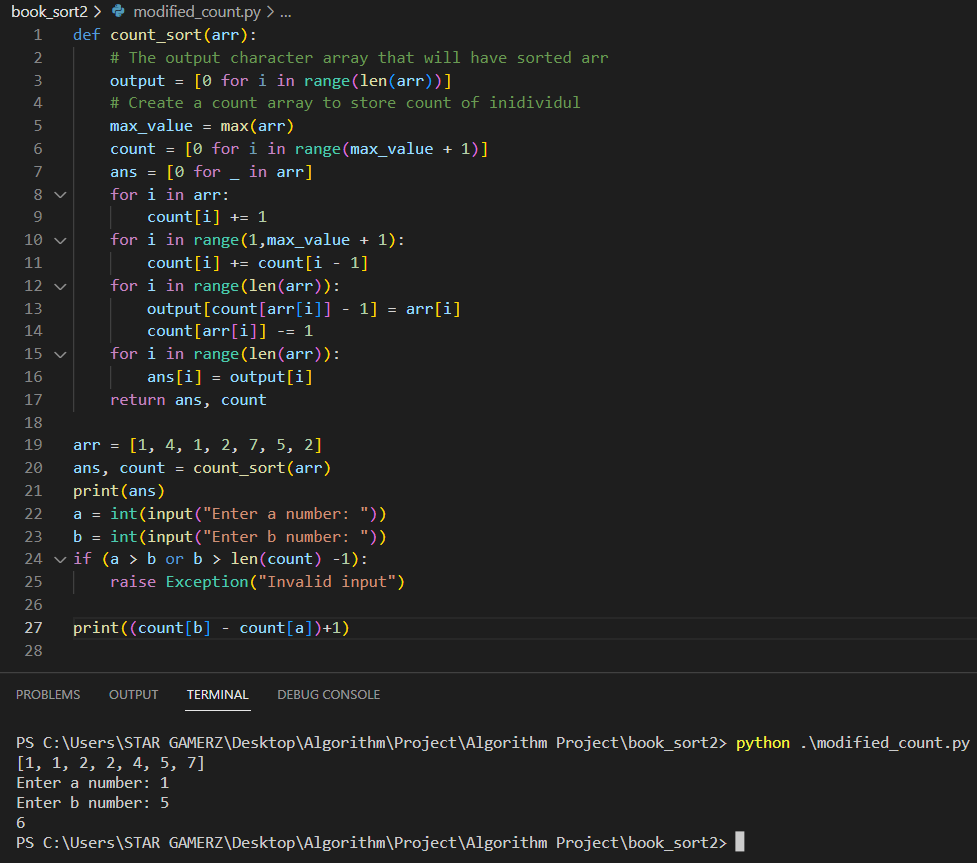


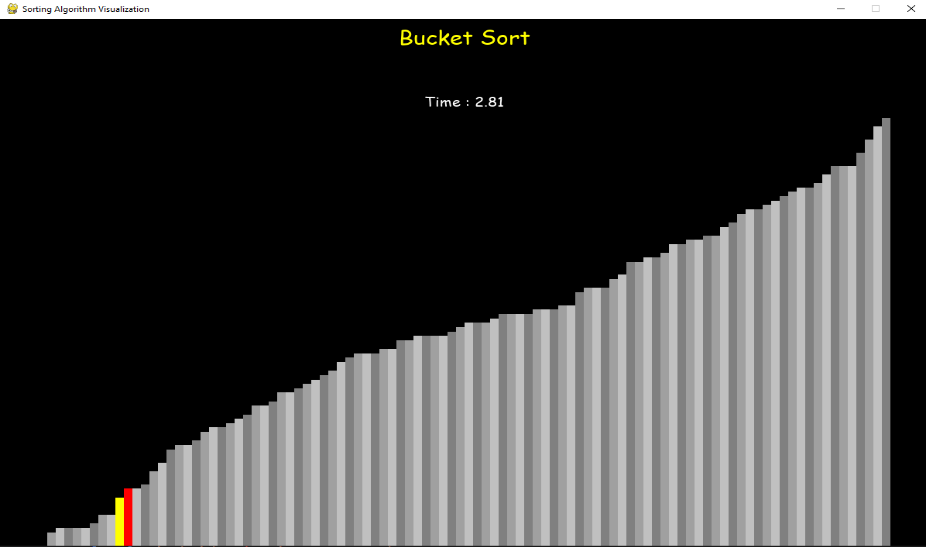












Conclusion:

|  |  |
| --- | --- |
| Sorting Algorithm | Time Calculated |
| Insertion sort | 38.67 |
| Bubble sort | 38.7 |
| Merge Sort | 0.85 |
| Heap Sort | 0.52 |
| Quick sort | 0.51 |
| Radix Sort | 1.23 |
| Bucket Sort | 2.81 |
| Courting sort | 1.65 |
| 7.4.5. from book | 0.52 |

Hence, we can conclude fastest sorting algorithm is quick sort and the slowest algorithm is bubble sort in our data set.