National University
of computer and emerging sciences

CS-2009

“Design & Analysis of Algorithms”
“PROJECT REPORT”

PROJECT TITLE:

“Sorting Visualization”

Group Members

Mohammad Usama 20K-0190

Fabiha Atique 20K-0369

Theory Instructor:

Dr. Farrukh Salim Shaikh

INTRODUCTION:

The Project name is **“Sorting Visualization”.** In this project we have implemented sorting algorithms in Python that you can visually see how a sorting algorithm will work and how sorting is performed. We have implemented 9 different algorithms. This will first ask the user which sorting algorithm you want to visualize and after that it will ask how many random array elements that you want to sort, and it will generate the text file of both random values and then sorted values and include which algorithm is performed. It will also display the running time of the algorithm that how much time it will take to sort that array

FEATURES:

- **Interactive user interface**
- **Animated view**
- **Fast & Efficient**

WHAT IS DIFFERENT IN OUR PROJECT:

We have approached a different approach in which we have implemented the sorting algorithms separately and then we have used python to visualize the sorting in this way the sorting visualization is easy to understand, and we can measure the running time of the algorithm.

TOOLS & Libraries:

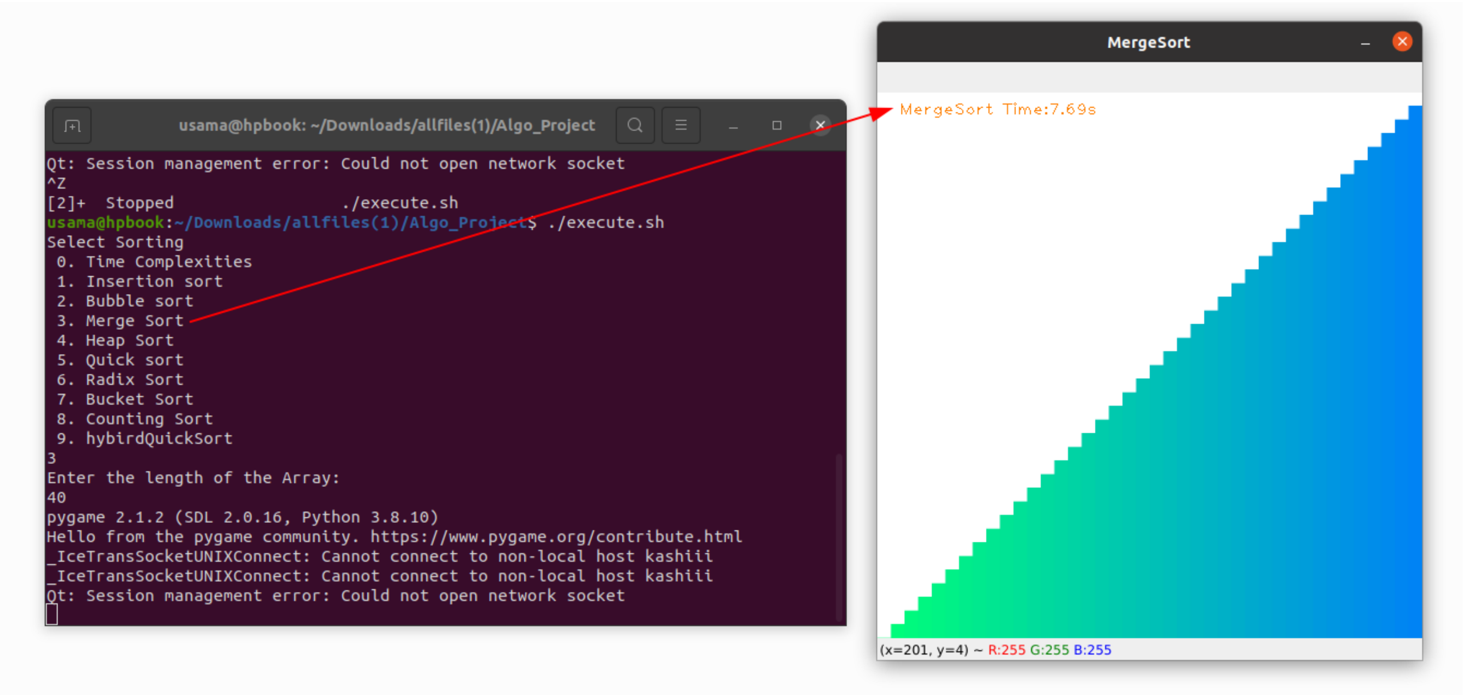
- **Windows/Ubuntu**
- **Python**
- **Shell Scripting**
- **Pygame**
- **Numpy**
- **CV2**
- **time**
- **random**

List of Sorting Algorithms:

- Insertion sort
- Bubble sort
- Merge Sort
- Heap Sort
- Quick sort
- Radix Sort
- Bucket Sort
- Courting sort
- HybirdSort
- • 8.2.4. from book

Screenshots:

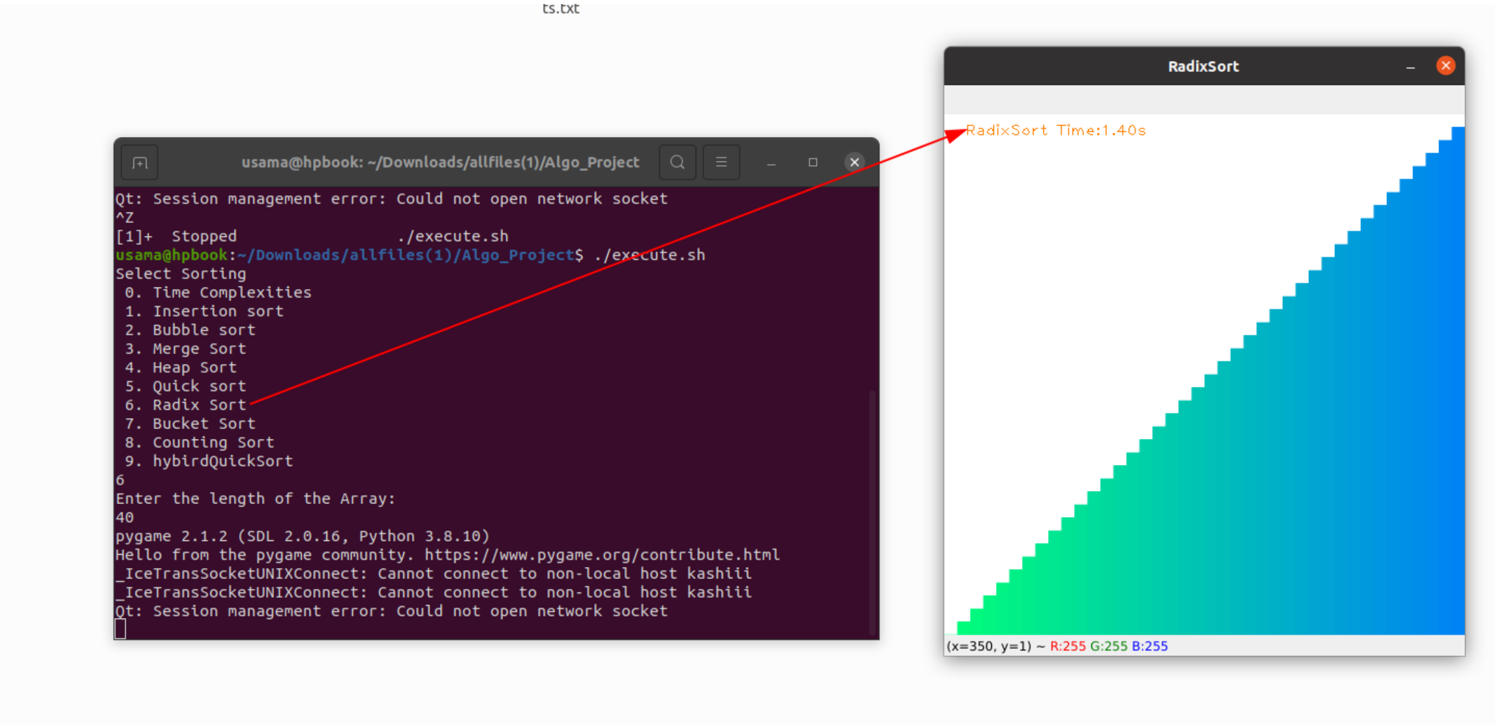
Merge Sort



usama@hpbook: ~/Downloads/allfiles(1)/Algo_Project
MergeSort
MergeSort Time:7.69s
(x=201, y=4) ~ R:255 G:255 B:255

```
Qt: Session management error: Could not open network socket
^Z
[2]+  Stopped                 ./execute.sh
usama@hpbook:~/Downloads/allfiles(1)/Algo_Project$ ./execute.sh
Select Sorting
 0. Time Complexities
 1. Insertion sort
 2. Bubble sort
 3. Merge Sort
 4. Heap Sort
 5. Quick sort
 6. Radix Sort
 7. Bucket Sort
 8. Counting Sort
 9. hybirdQuickSort
3
Enter the length of the Array:
40
pygame 2.1.2 (SDL 2.0.16, Python 3.8.10)
Hello from the pygame community. https://www.pygame.org/contribute.html
_IceTransSocketUNIXConnect: Cannot connect to non-local host kashiii
_IceTransSocketUNIXConnect: Cannot connect to non-local host kashiii
Qt: Session management error: Could not open network socket
```

Radix Sort

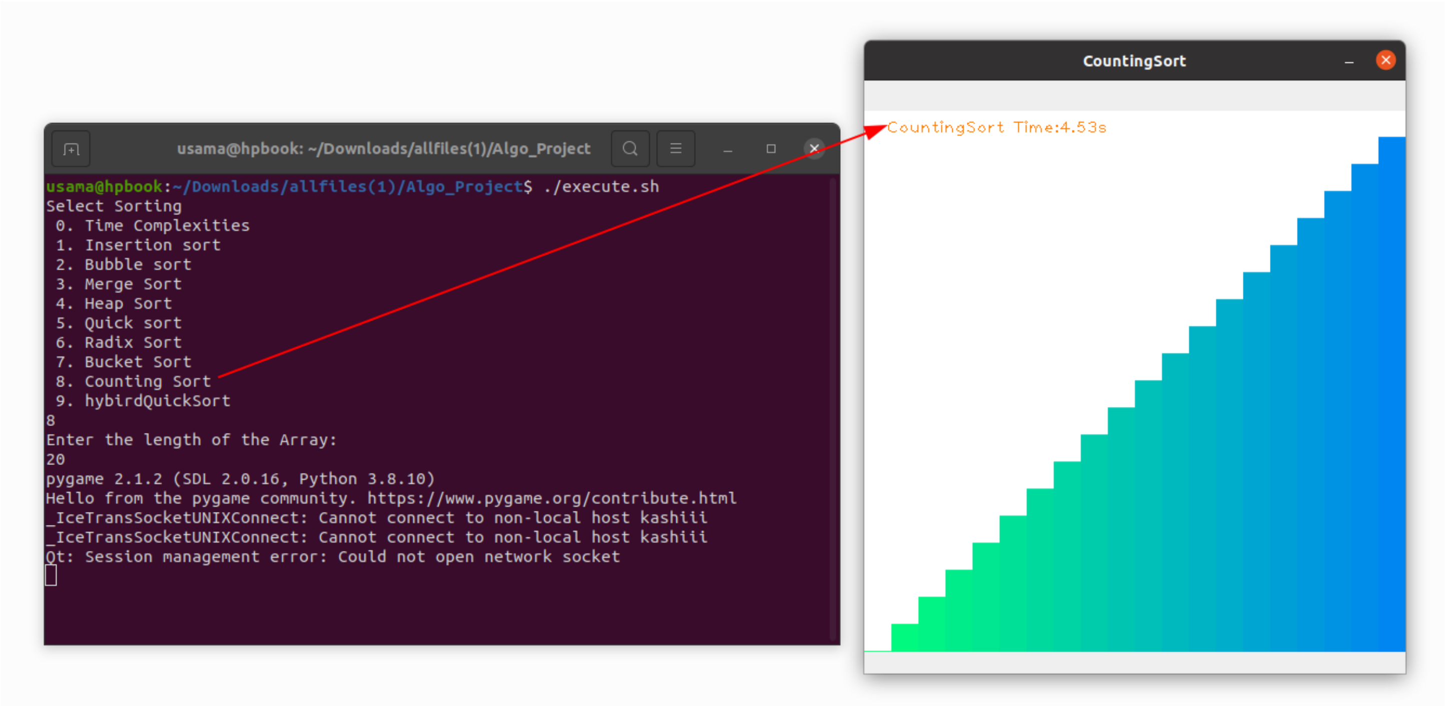


```
Qt: Session management error: Could not open network socket
^Z
[1]+  Stopped                 ./execute.sh
usama@hpbook:~/Downloads/allfiles(1)/Algo_Project$ ./execute.sh
Select Sorting
 0. Time Complexities
 1. Insertion sort
 2. Bubble sort
 3. Merge Sort
 4. Heap Sort
 5. Quick sort
 6. Radix Sort
 7. Bucket Sort
 8. Counting Sort
 9. hybirdQuickSort
6
Enter the length of the Array:
40
pygame 2.1.2 (SDL 2.0.16, Python 3.8.10)
Hello from the pygame community. https://www.pygame.org/contribute.html
_IceTransSocketUNIXConnect: Cannot connect to non-local host kashiii
_IceTransSocketUNIXConnect: Cannot connect to non-local host kashiii
Qt: Session management error: Could not open network socket
```

```
RadixSort Time:1.40s
```

```
(x=350, y=1) ~ R:255 G:255 B:255
```

Counting Sort



```
usama@hpbook:~/Downloads/allfiles(1)/Algo_Project$ ./execute.sh
Select Sorting
 0. Time Complexities
 1. Insertion sort
 2. Bubble sort
 3. Merge Sort
 4. Heap Sort
 5. Quick sort
 6. Radix Sort
 7. Bucket Sort
 8. Counting Sort
 9. hybirdQuickSort
8
Enter the length of the Array:
20
pygame 2.1.2 (SDL 2.0.16, Python 3.8.10)
Hello from the pygame community. https://www.pygame.org/contribute.html
_IceTransSocketUNIXConnect: Cannot connect to non-local host kashiii
_IceTransSocketUNIXConnect: Cannot connect to non-local host kashiii
Qt: Session management error: Could not open network socket
```

```
CountingSort Time:4.53s
```

CONCLUSION:

Finally, The benefit of this project is that everyone can understand the working of sorting algorithms visually and Python makes it more easy. In future more sorting algorithms will be added and these types of visualizations can be used in universities to teach new students how sorting algorithms will sort that array values because too many sorting algorithms will confuse them but if we can show them visualization then they will be clearer.