**Project Report**

**Group members:**

**20k-0477**

**20k-0385**

**Abstract:**

Nowadays sorting algorithms are widely used in computer software. For example, if you open file explorer on your PC, you may see files sorted in different ways. The aim of this project is to show implementation of various sorting algorithms and compare their time complexities and how much time they take to sort an array and how do they sort it by visualization.

**Introduction:**

The main goal of the project was to create a program which would serve as a tool for understanding how most sorting algorithms work. Our project is about implementation of sorting algorithms with their time complexities which includes

* Insertion sort
* Bubble sort
* Merge Sort
* Heap Sort
* Quick sort
* Radix Sort
* Bucket Sort
* Courting sort
* 7.4.5. from book
* 8.2.4. from book

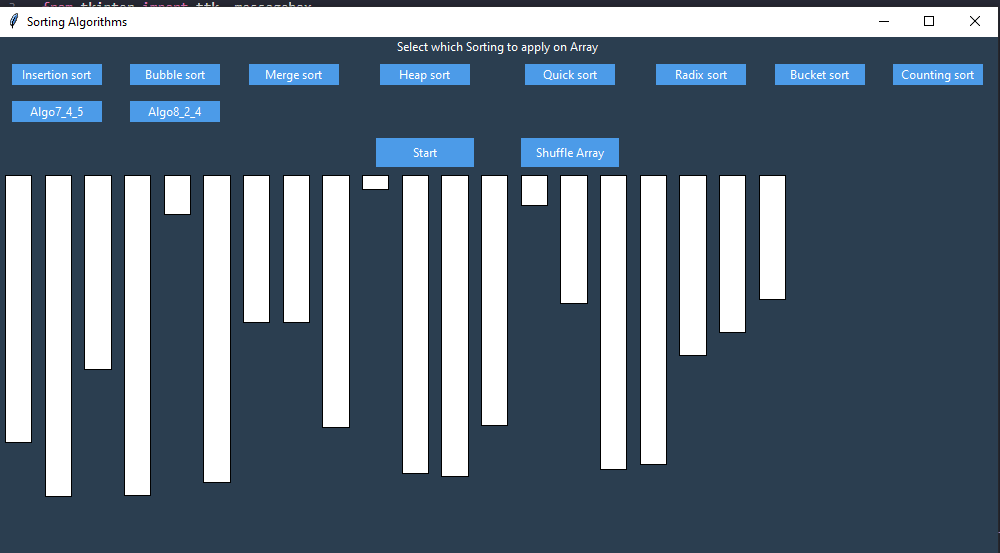
We will show by visualization that how each algorithm sorts an array and what is the position of elements while they are being sorted and how much time it takes to sort by each algorithm and compare them.

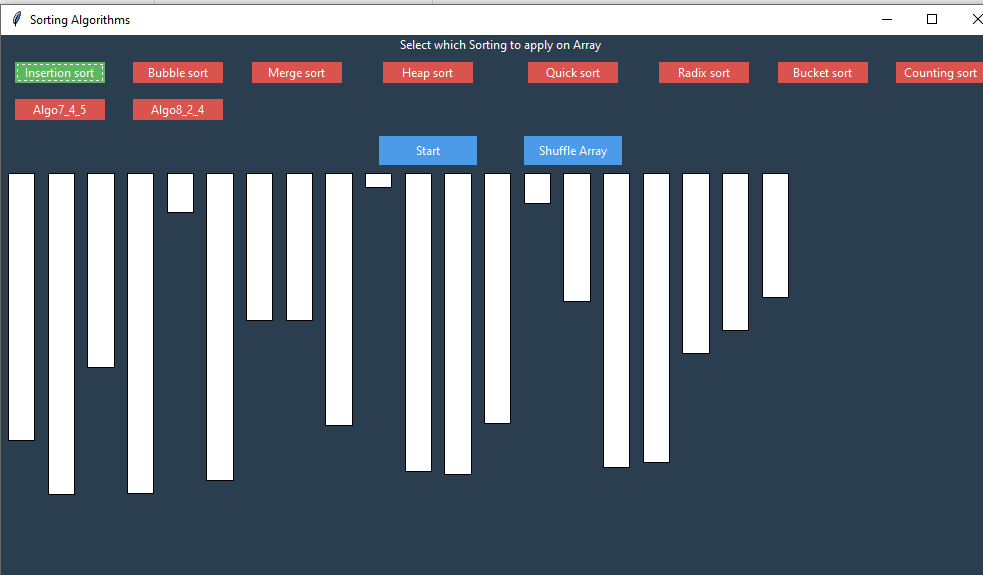
**Programming Design:**

We used **python** for making our project. In addition, we have used tkinter package for showing visualization. The tkinter package (“Tk interface”) is the standard Python interface to the Tcl/Tk GUI toolkit. Both Tk and tkinter are available on most Unix platforms, including macOS, as well as on Windows systems. We saw bars as our numbers for visualization of array elements. As the algorithm continues working, the bars position change showing how the array elements are changing.

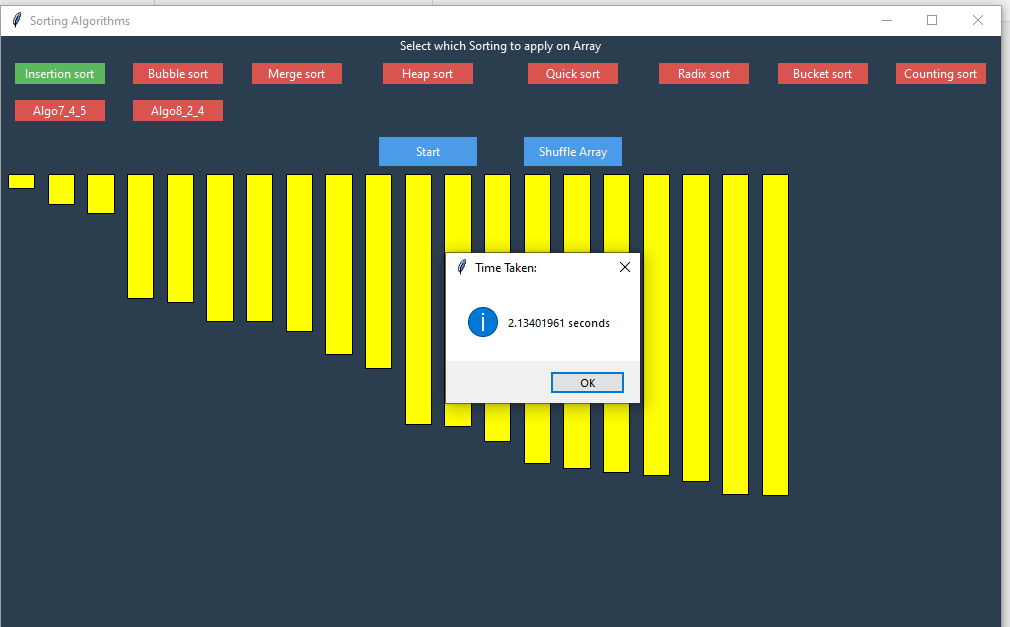
**Results:**

Once the program is compiled, a txt file is generated with random numbers and are stored in our array. A menu appears asking the user which algorithm he wants to see. Once he chooses and clicks on start, the algorithm starts working and arrays start getting sorted, and the bars starts changing position to show visualization and how the array elements are getting sorted.





After sorted





While sorting