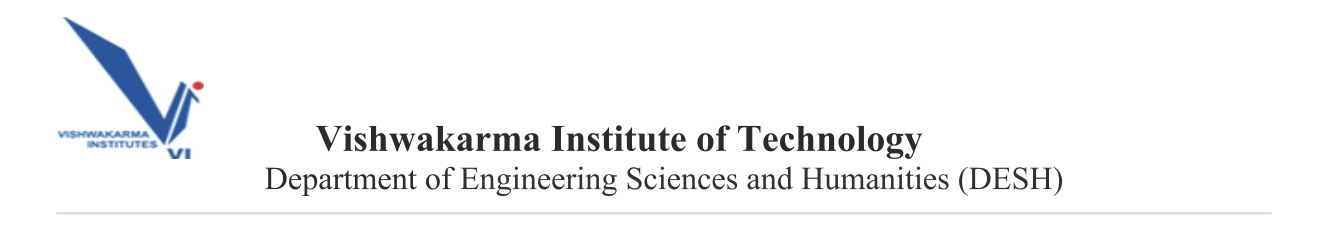
**Visualizing Sorting Algorithm**

Kunal Patil (12011121), Mandar Patil (12010845), Maruti Patil (12011071) , Niraj Patil (12011336) , Omkar Patil (12010795) , Parth Patil (12010088)

Department of Engineering, Sciences and Humanities (DESH)

*Abstract*— *Many of the people while starting the journey of programing face sorting algorithm difficult to understand. That’s the reason why most of them leave it. So, we have created a Visualization of these sorting algorithm using python and tkinter module.*

Keywords— bubble Sort, education, insertion Sort, quick Sort, tkinter, visualize

# Introduction

A Sorting Algorithm is an algorithm that puts elements of a list in a certain order, such as alphabetical, highest-to-lowest value or shortest-to-longest distance. To visualize an algorithm, we don’t merely fit data to a chart; there is no primary dataset. Instead, there are logical rules that describe behavior. The output of any sorting algorithm must satisfy two conditions:

* The output is in nondecreasing order (each element is no smaller than the previous element according to the desired [total order](https://en.wikipedia.org/wiki/Total_order)s);
* The output is [permutation](https://en.wikipedia.org/wiki/Permutation) (a reordering, yet retaining all of the original elements) of the input.

We implemented GUI (Graphical User Interface) using Tkinter package in python. There are multiple ways to sort algorithms:

1. Quick Sort
2. [Bubble Sort](https://www.interviewbit.com/tutorial/bubble-sort/)
3. [Merge Sort](https://www.interviewbit.com/tutorial/merge-sort-algorithm/)
4. [Insertion Sort](https://www.interviewbit.com/tutorial/insertion-sort-algorithm/)
5. [Selection Sort](https://www.interviewbit.com/tutorial/selection-sort/)
6. Heap Sort
7. Radix Sort
8. Bucket Sort

Here, we will be focusing on the major ones which include **Bubble Sort, Insertions Sort & Quick Sort**.

# Implementation details

*A. System Architecture*

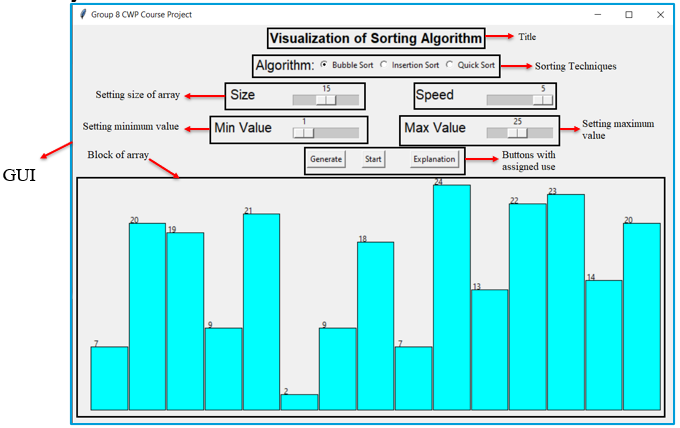
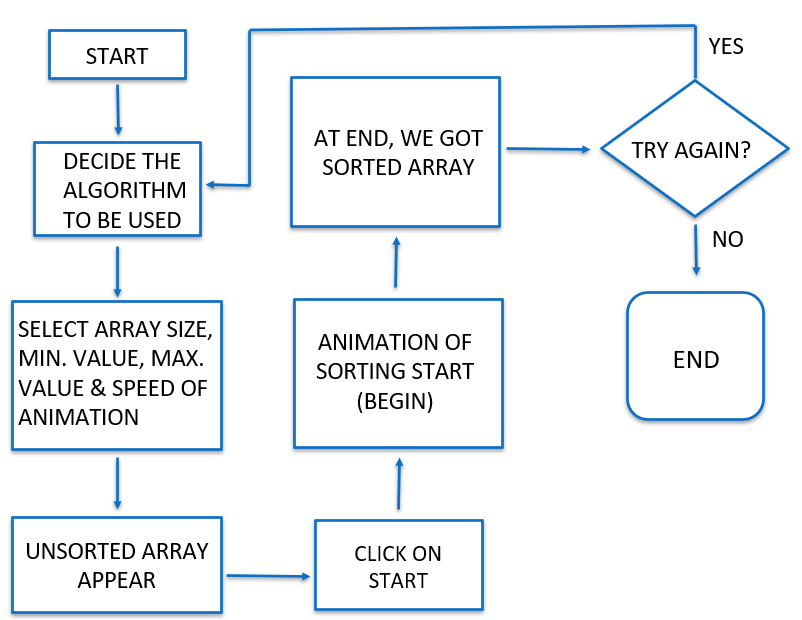
**

Fig.: 1 Visualization of Sorting Algorithm

The System Architecture consist the whole GUI that we had made for our project. As shown in the above Fig.1, the name of GUI is given as "Group 8 CWP Project" and title as "Visualization of Sorting Algorithm". We have provided 3 types of sorting techniques to the user, the user can choose one from the given, namely as Bubble, insertion and quick sort. The user can set size of array ranging from 5 to 25 and also the min. value ranging from 0 to 50 and max. value ranging from 5 to 50. We have provided 2 buttons "Generate" and "Start", Generate: generates random no. of blocks and Start: starts to sort the array according to the selected sorting technique.

*B. Flowchart*



*C. Algorithm*

1. Bubble Sort

When we consider a data collection to be sorted using Bubble Sort, this implies that algorithm will require maximum n-1 passes and [n(n-1)]/2 comparisons for a data collection to get sorted. The algorithm sorts a data collection in such a way that at 1st pass the greatest element in the collection reaches at end position, similarly for pass 2, 2nd greatest element reaches at 2nd last position in the collection and likewise other elements are sorted. The movement of elements is done by pair to pair checking of elements and if they are in undesired order they are shuffled. It is more suitable for small lists.

|  |  |  |
| --- | --- | --- |
| Best case | Average case | Worst case |
| N | N2 | N2 |

2. Insertion Sort

In case of Insertion Sort, the algorithm builds final sorting list by transferring one element at a time and the algorithm requires maximum n-1 passes and (n-1)/4n comparisons for a data collection to get sorted. It considers the 1st element of the data collection as sorted, then all other elements are one by one compared with that element and every element is inserted as per its value in the sorted list. If compared to bubble sort, Insertion sort is twice as fast as it.

|  |  |  |
| --- | --- | --- |
| Best case | Average case | Worst case |
| N | N2 | N2 |

3. Quick Sort

Quick Sort also known as Divide and conquer algorithm as it picks an element as pivot and partitions the given data collection around the picked pivot. Inside Quick Sort major processing is done by Partition function, where we keep the track of last smallest element (w.r.t. Pivot element) of data collection with a variable and we keep the track of first greatest element (w.r.t. Pivot element) of data collection by using another variable. And after every pass Pivot element is brought to its sorted position i.e., all, elements on its left are smaller than it and all elements on right are greater. Further the data collection is divided into two fragments and again the quick sort algorithm is applied i.e., it calls itself and therefore it is a recursive algorithm.

|  |  |  |
| --- | --- | --- |
| Best case | Average case | Worst case |
| NLogN | NLogN | N2 |

# Results and discussion

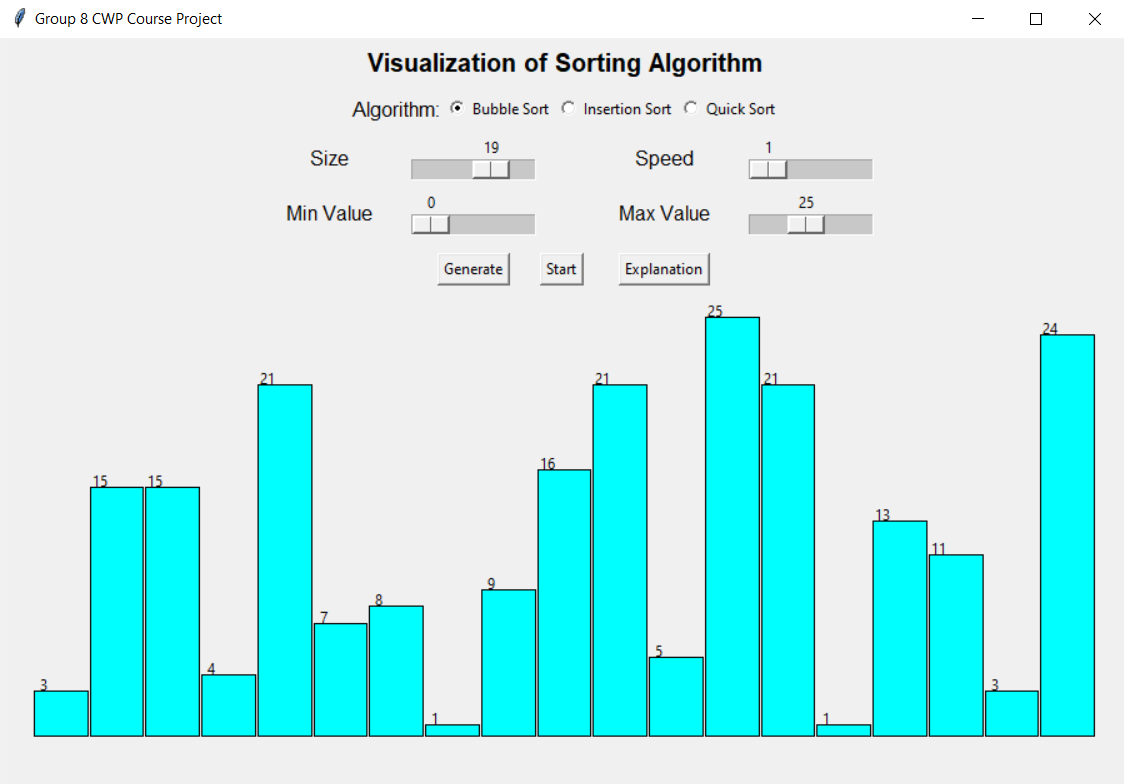


Fig.: 2 Generating random no. of blocks

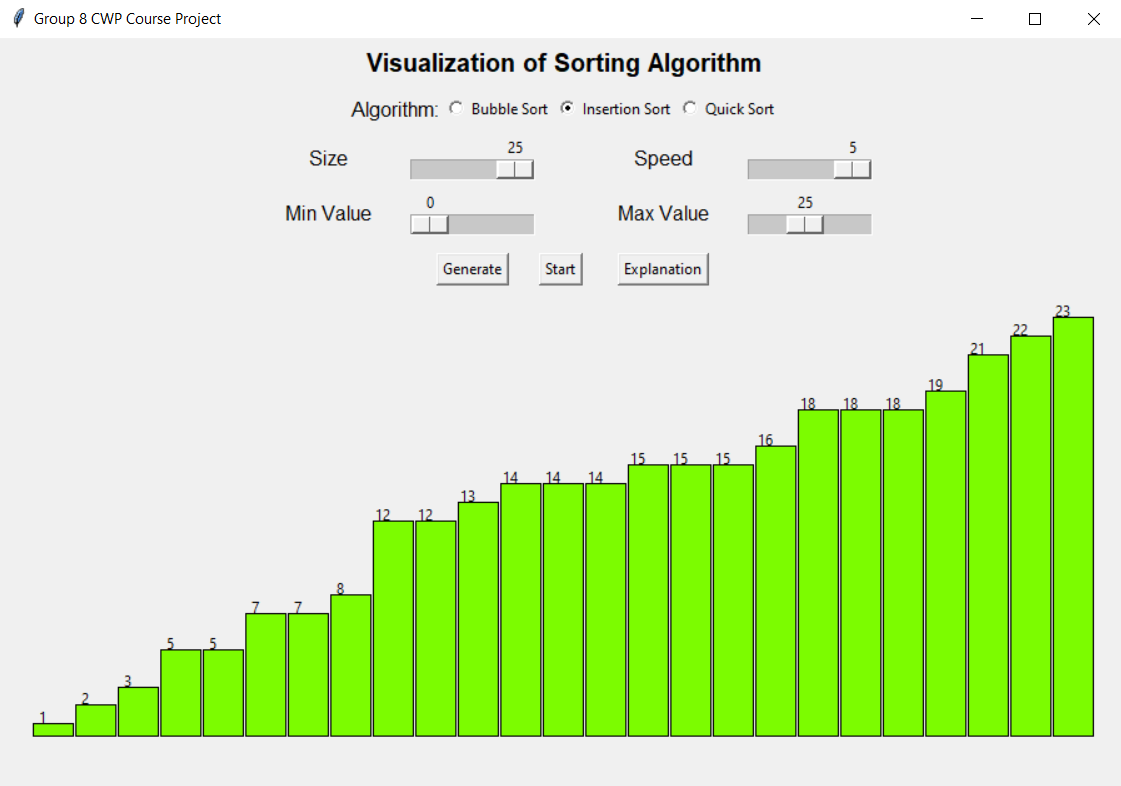


Fig.: 3 Results after sorting is complete

As shown in the above Fig.2, a random number of unsorted blocks are created, by clicking the generate button, after setting the max and min values first. And after selecting the required sorting technique it starts to sort the array according to the given algorithm. Then we get our final result as shown in the Fig.3, the sorted array, which is sorted by the code itself.

# Conclusion and Future Scope

Ultimately the code arranges the elements according to the sorting method chosen. The first iteration will find the first biggest element, the second iteration will second biggest element and so on. These different sorting methods give the user an easy hand to convert the random value bars into ascending or descending graphs.

We can also sort the elements right after the user inputs the values.

##### References

[1] https[://www.geeksforgeeks.org/python-gui-tkinter/](https://www.geeksforgeeks.org/python-gui-tkinter/)

[2] code with Harry’s tkinter tutorial (<https://youtube.com/playlist?list=plu0w_9lii9ajlcqrcj4poeihkukf_otza>)

[3] <https://youtu.be/xglrx34kvj8>

[4] <https://www.geeksforgeeks.org/bubble-sort/>

[5]https[://www.geeksforgeeks.org/insertion-sort/](https://www.geeksforgeeks.org/insertion-sort/)

[6] <https://www.geeksforgeeks.org/quick-sort/>