**ABSTRACT**:

Sorting Visualizer is an application that allows user to select a file and the sorting algorithm and can see step by step visualization of how the elements are being sorted in real time fashion. The sorting visualizer will not only entertain the illustrations but also tell iterations, number of steps and the resulted arrays after a single manipulation.

**INTRODUCTION**:

The theme sorting visualizer has carried forward is to have common understanding to common laymen that how different sorting algorithms behave when given same input, so we can have an idea about what sorting algorithm is best in specific cases. The sorting visualizer is an application which is enabled to ask user’s choice input file and then have a walkthrough of each and every single iteration of the algorithm on which it will be made available. The main crux and heart of the sorting visualizer lies in the fact that it will entertain users input on the sorting algorithm chosen.

**PROGRAMMING DESIGN:**

The program is developed on python 3.10 using standard GUI library “Tkinter” for visualizations. The program starts on a single frame having three canvases, labeled as:

1. Heading (Application name and the designers name)
2. This canvas will be holding multiple options: Making user to select file from his system, select the algorithm and asking the user to select sorting speed ranging from (0 – 2x).
3. Visualizer that will sort the algorithm step by step

Once the sorting has been completed, the user will be carried forward to a second frame that will be holding the number of iterations, resulting arrays after each single manipulation also with time complexity of that sorting algorithm in which the inputs are given. Once the results have been displayed, you may choose another file and sorting algorithm until users wish.

**EXPERIMENTAL SETUP:**

The Sorting Visualizer will ask a file from the user which must be given in .txt format, either space separated or comma separated, input values. Once the file is selected from the system you can select sorting algorithm from any of the following algorithms:

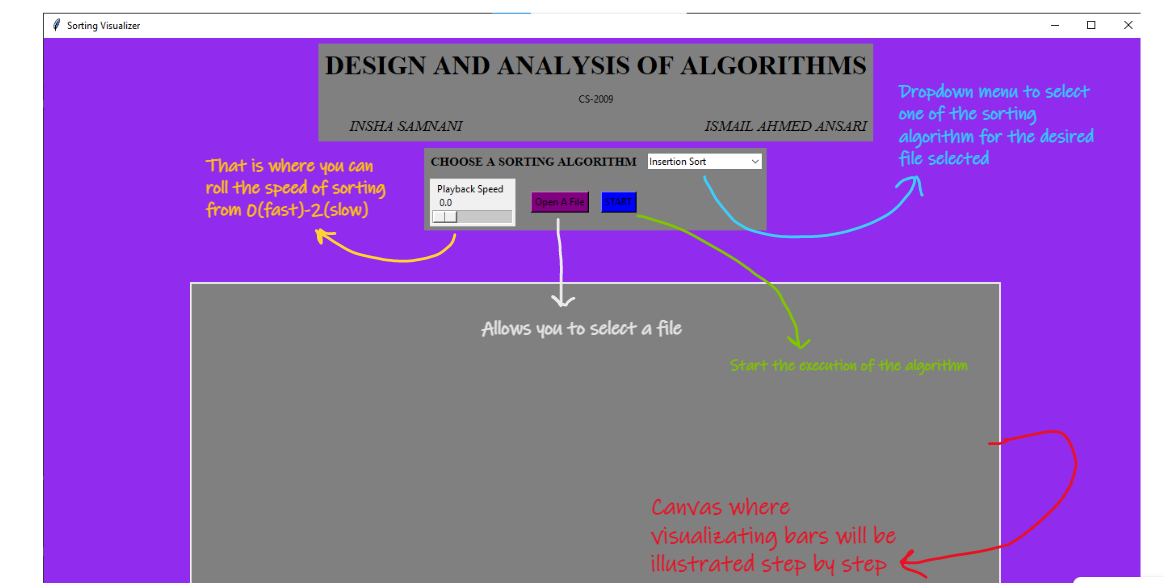
1. Insertion sort
2. Bubble sort
3. Merge Sort
4. Heap Sort
5. Quick sort
6. Radix Sort
7. Bucket Sort
8. Courting sort
9. 7.4.5. from book
10. 8.2.4. from book

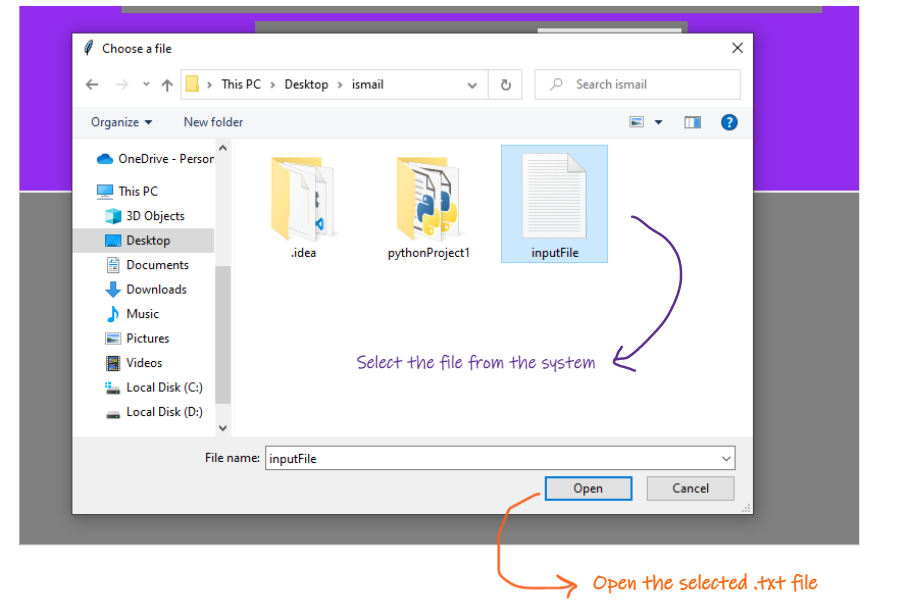
Once the algorithm selected you can select the speed timeframe by adjusting the sidebar that how fast algorithm visualization and iterations to be displayed on a speed of (0-2x), where 0x indicates the faster execution and you can go as low speed of 2x. Once its all done, on click of the start button the algorithm will begin to execute. After the successful visualization another window will be presented having:

1. Iterations and Sub Iterations
2. Resulting Arrays after Comparison/Swapping manipulations
3. Outplace Sorting algorithms using another array manipulations
4. Time Complexity of the algorithm
5. Comparisons/Swaps/Iterations/Calls/Total count
6. Space Complexity of algorithm, visualization and dry run allocations

You can run this as many times one after another and see the resulting frames accordingly.

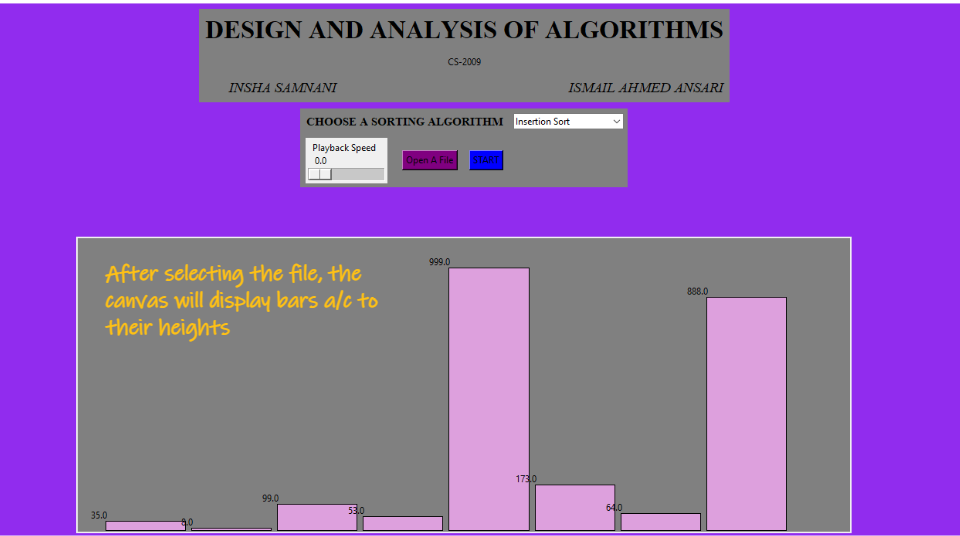
**RESULTS:**

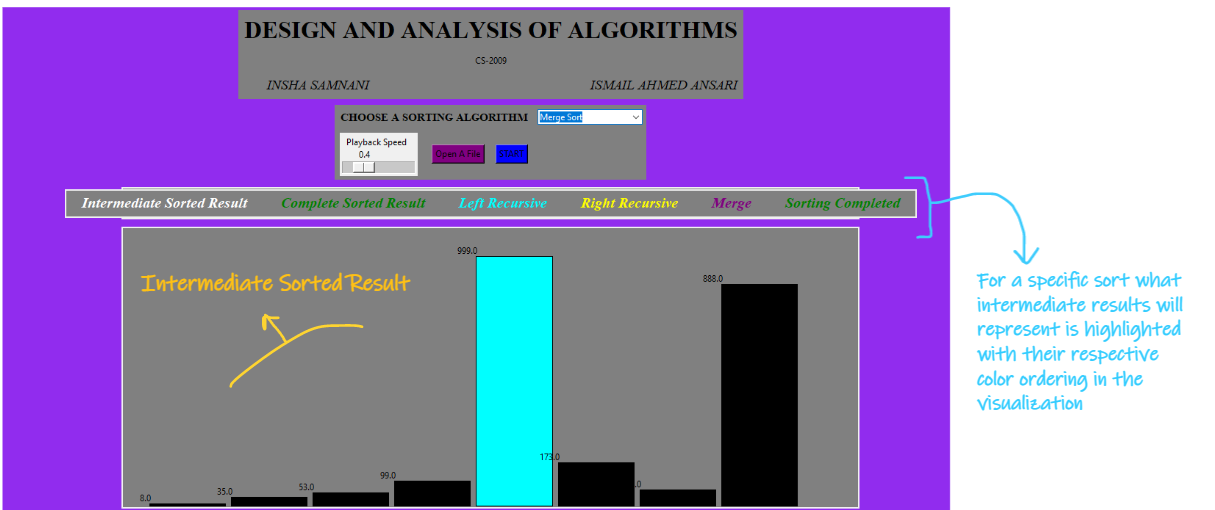


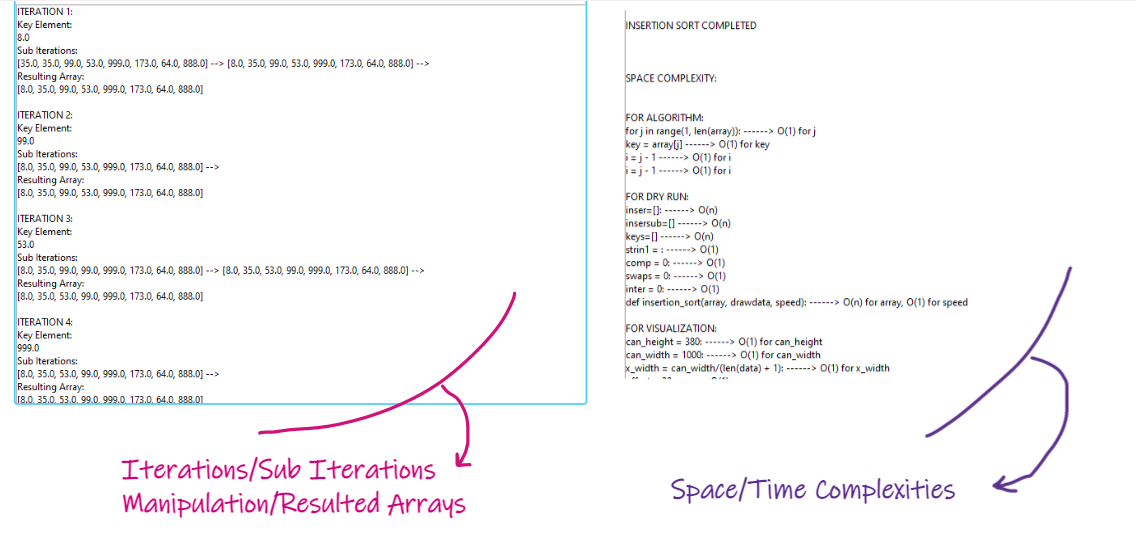


Graphical user interface, text, application

Description automatically generated







The process can be repeated if the user wishes to cancel the program on its own.

**CONCLUSION**:

The sorting visualizer is perfect for a person eager to learn how different sorting algorithms behave when executed on different input sets and to get a better view of each visualization of every algorithm and get to know the time/space related complexity ahead of the input size. Some improvements can be made to make the application more stable by allowing different file formats, frames deallocation, and using file system for storing the results for later use. But for now, the requirements to what demanded have been fulfilled.

**REFERENCES**:

The reference taken is from a GitHub repository:

https://github.com/JordinaGR/sorting-algs-visualization/tree/master/algs