NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES – KARACHI CAMPUS

DESIGN AND ANALYSIS OF ALGORITHMS
SECTION: BCS-5E

SORTING ALGORITHMS VISUALIZER

PROJECT REPORT

GROUP MEMBERS:

MUHAMMAD WARZAN 20K-1649
AWWAB SABIR 20K-1615

SORTING ALGORITHM VISUALIZER

ABSTRACT

One of the simplest areas of research in computer science is sorting algorithms. The intention is to make searching, inserting, and deleting records simpler. The time and space complexity of five sort algorithms like bubble, choose, insert, merger, and quick was outlined. Also discovered were two categories of O(n2) and O(n log n). Based on the trials, certain conclusions on the extent and degree of randomness of the input sequence were made. Insertion sort or selection sort function effectively with tiny record sizes. Insertion sort or bubble sort work effectively in an ordered sequence. Quick sort or merge sort works effectively when the number of records is huge. Various applications might choose the best sort algorithm based on these guidelines.

INTRODUCTION

The effectiveness of animated sorting algorithms for teaching was examined in this study. The construction of a web-based animation programme allowed the visualisation of several sorting techniques. A bar graph is used to visualise the data, following which an algorithm and data sorting may be used. The finished animation is then presented, with the viewer controlling the pace. This study focuses on how algorithms are taught in computer science curriculum. A presentation and a survey were both part of the experiment. In this study, these results and responses are compiled and examined with findings from past studies.

Sorting algorithm specifies the way to arrange data in a particular order. Most common orders are in numerical or lexicographical order. The importance of sorting lies in the fact that data searching can be optimized to a very high level, if data is stored in a sorted manner.

Keeping in mind the view and importance of algorithms we have developed a project consisting of sorting algorithms visualizations in a user friendly interface. Below is the list of algorithms:

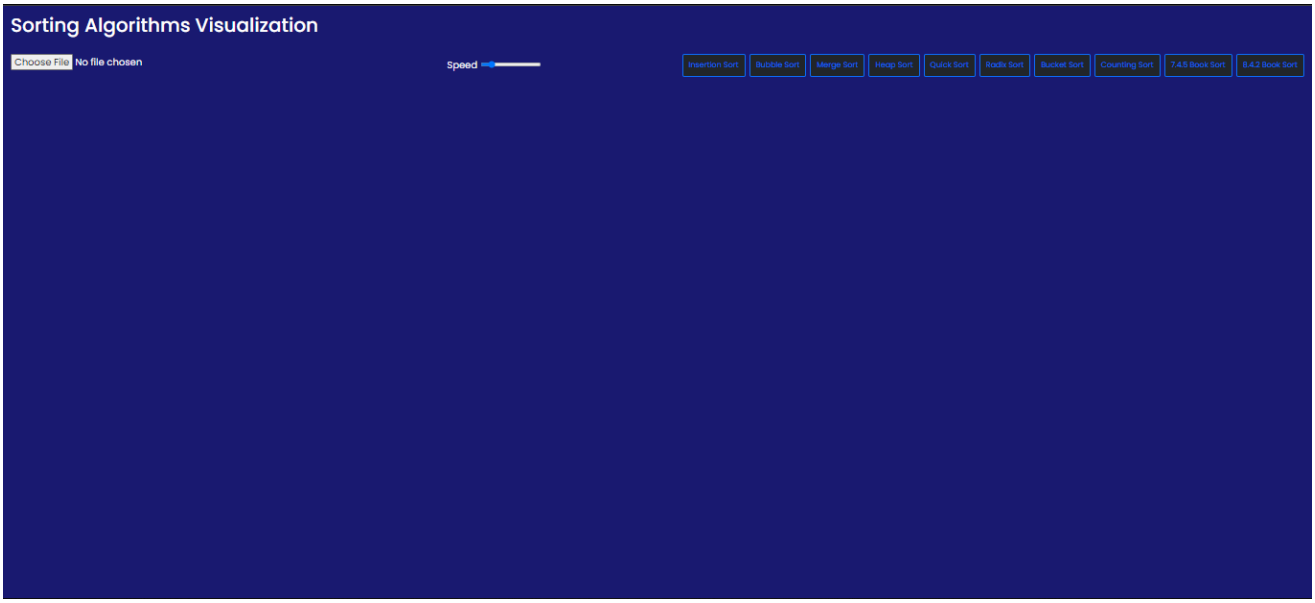
- Bubble Sort
- Insertion Sort
- Merge Sort
- Quick Sort
- Heap Sort
- Radix Sort
- Counting Sort
- Bucket Sort
- 7.4.5 from book
- 8.2.4 from book

We uses HTML5 (Hypertext Markup Text Language) JavaScript, and CSS (Cascading Style Sheets) for the website's layout.

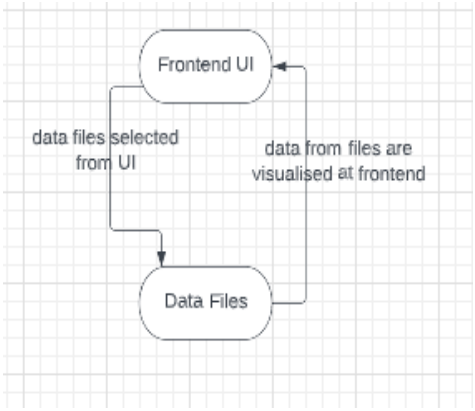
DESIGN

The user has the option to choose a file and pick the sorting method he prefers. The layout is really user-friendly and straightforward. For the layout of the website, we selected VS Code as our editor along with HTML5 (Hypertext Markup Text Language), JavaScript, and CSS (Cascading Style Sheets).

Initially, when the application loads in the browser, this is the user interface that you would see;



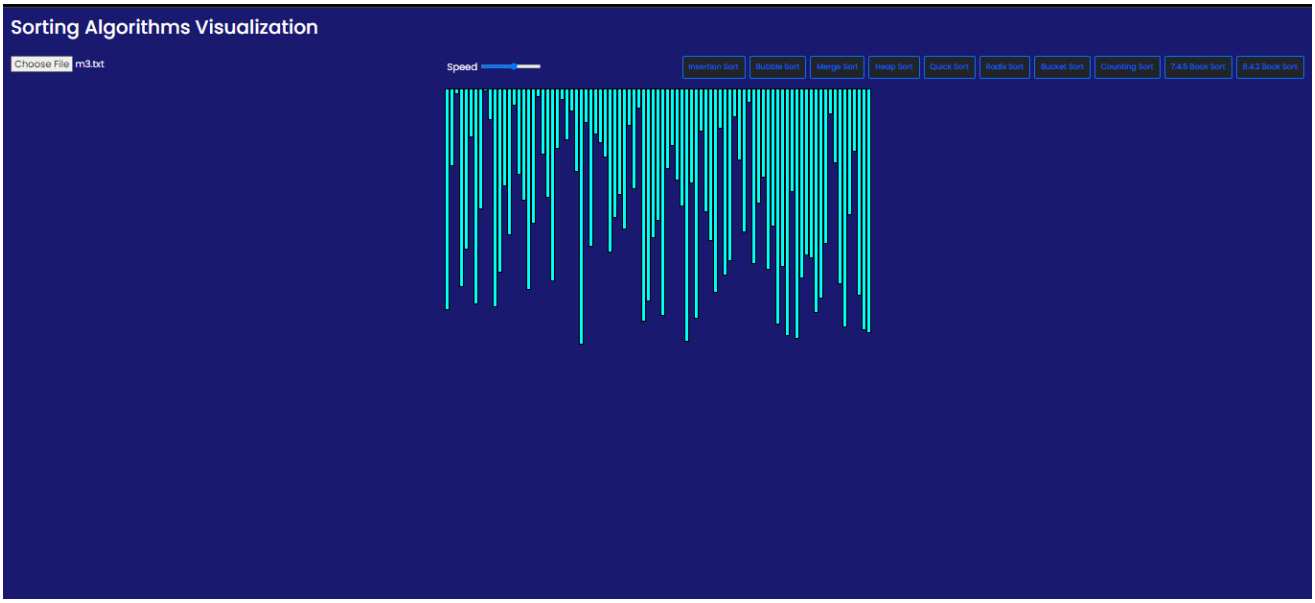
Basic flow of our application;



The user is first able to choose any text file from the file explorer that contains the desired array of various sizes. After that, the browser loads the specified text file and displays it as a bar graph. The user may also alter the algorithm's speed with the use of a speed slider before choosing the sorting algorithm they want to employ to order the loaded array. We also reported the time complexity, space complexity, and the amount of time a method took in seconds once the sorting visualisation was finished.
The use of HTML5 (Hypertext Markup Language 5), JavaScript, and CSS combine to form this project's implementation (Cascading Style Sheets). There is only one project file which is an HTML file and contains the code.

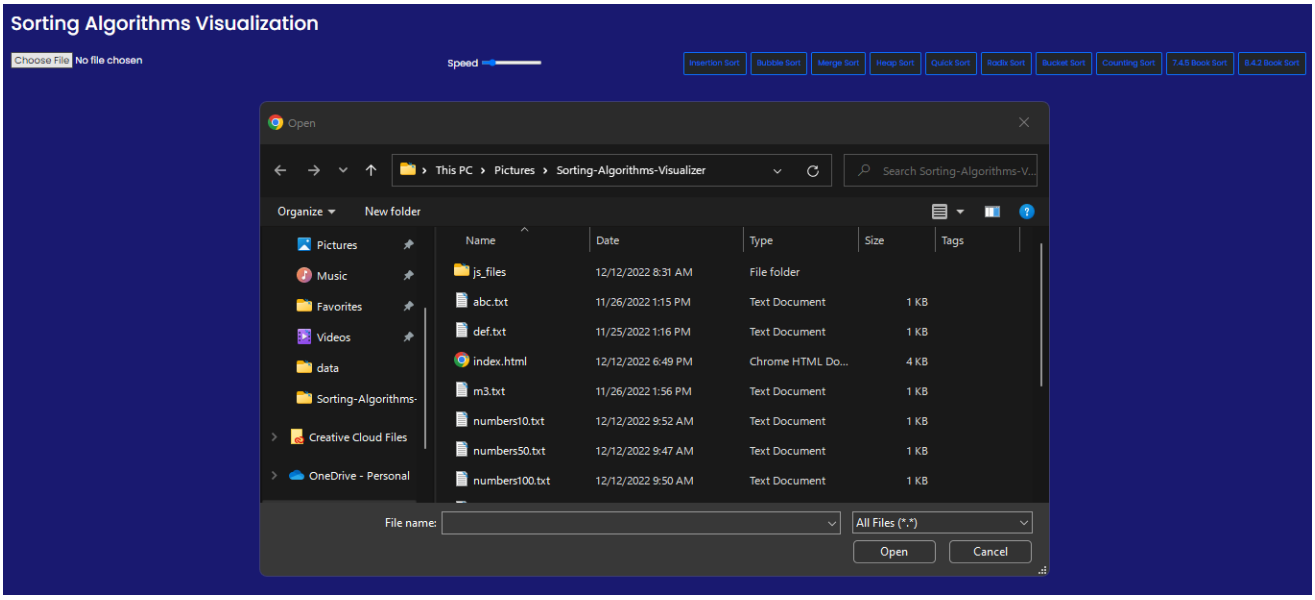
EXPERIMENTAL SETUP

Using random files of different sizes (10, 50, 100, 200, 300, 400, 500, 1000) the sorting methods were experimented with and tested for validity. Each sorting algorithm performed as expected and produced accurate results. Some took up more space, while others were slower because of their time-complexity.

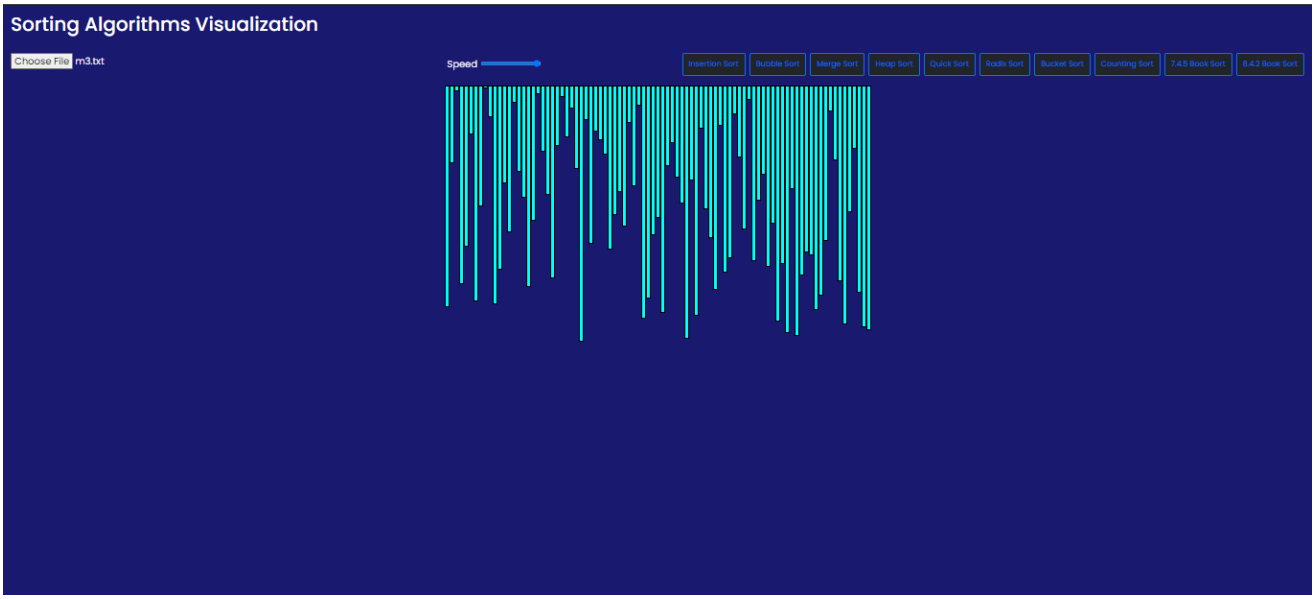


RESULTS AND DISCUSION

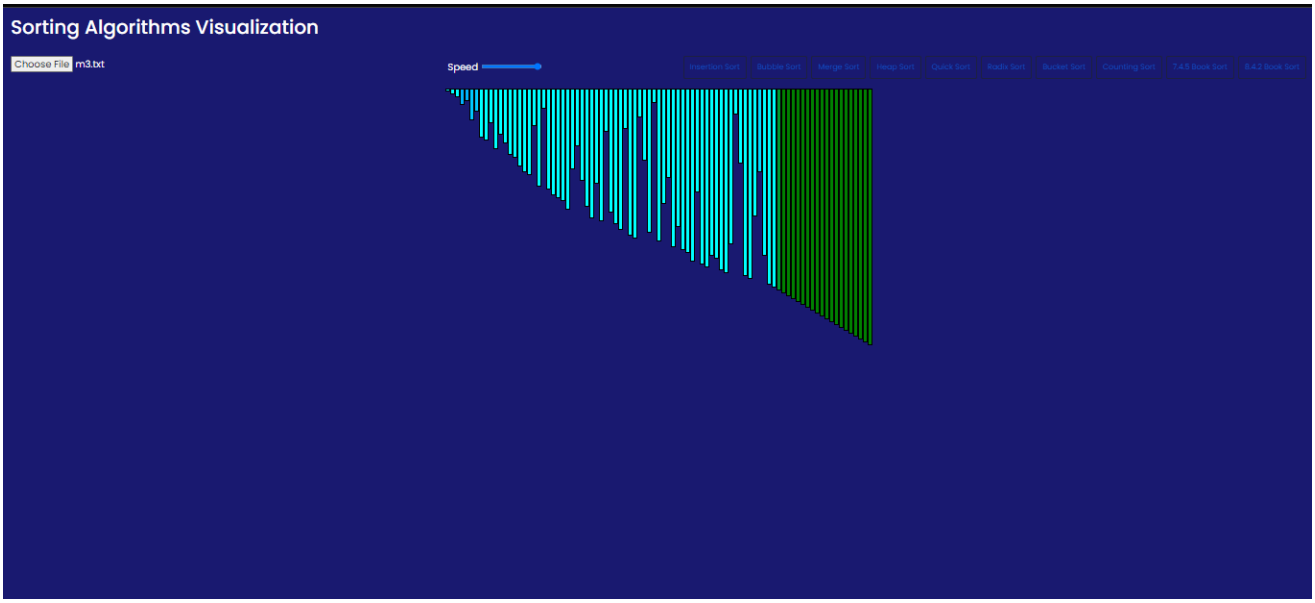
While choosing the desired file from the File Explorer;



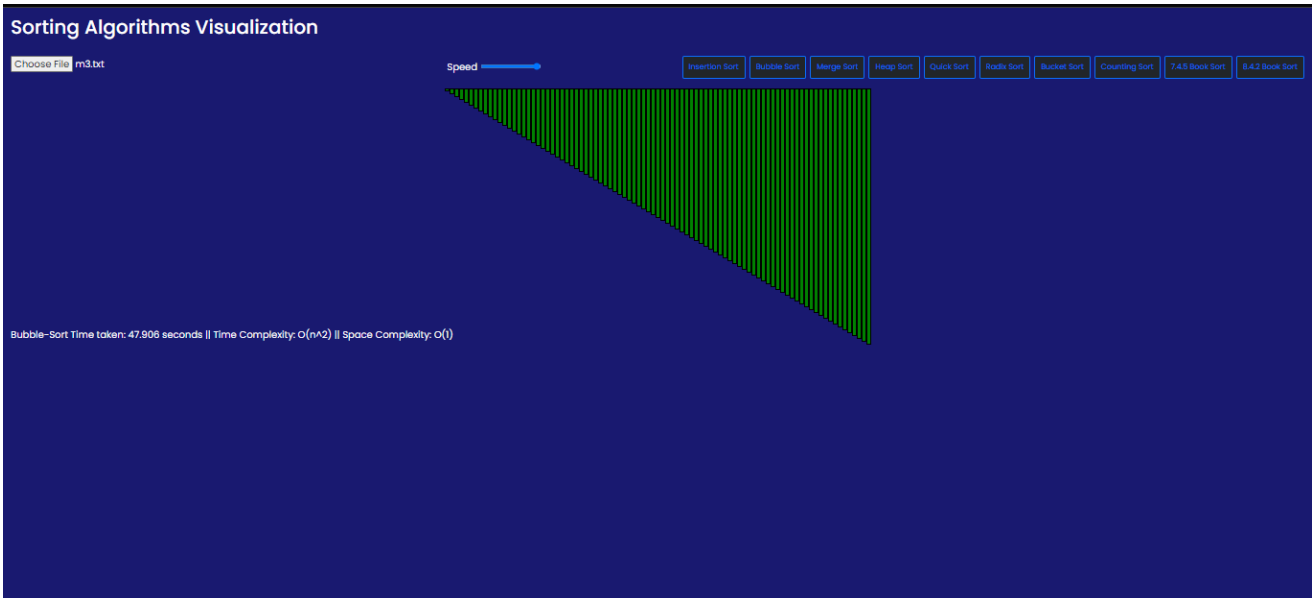
Before started sorting visualization;



While during sorting visualization;



After executing sorting visualization get the sorted array plus the time lapse and time/space complexities;



CONCLUSION

Rearranging data into a predetermined order, such as ascending or descending, is the process of sorting. As long as it is a linear, or complete ordering, meaning that any two items may be ordered, it doesn't really matter how the elements are arranged. As the main goal of this semester project, we tried our best and created the teaching support application which visualizes the most known sorting algorithms. User may run his/her desired sorting on a random or custom array by choosing a text file from File Explorer. I tried to create high-quality software with a user-friendly and easy-to-use interface, which could be used by other students.

REFERENCES

[1] Algorithms in Java, Parts 1-4, 3rd edition by Robert Sedgewick. Addison Wesley, 2003.
[2] Programming Pearls by Jon Bentley. Addison Wesley, 1986.
[3] Quicksort is Optimal by Robert Sedgewick and Jon Bentley, Knuthfest, Stanford University, January, 2002.
[4] Dual Pivot Quicksort: Code and Discussion.
[5] Bubble-sort with Hungarian ("Csángó") folk dance YouTube video, created at Sapientia University, Tirgu Mures (Marosvásárhely), Romania.
[6] Select-sort with Gypsy folk dance YouTube video, created at Sapientia University, Tirgu Mures (Marosvásárhely), Romania.
[7] Sorting Out Sorting, Ronald M. Baecker with the assistance of David Sherman, 30 minute color sound film, Dynamic Graphics Project, University of Toronto, 1981. Excerpted and reprinted in SIGGRAPH Video Review 7, 1983. Distributed by Morgan Kaufmann, Publishers.