A REPORT ON
SORTING VISUALIZER
ALONG WITH THE ALERT FEATURE

PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY

(Computer Science & Engineering)

SUBMITTED BY

KESHAV KUMAR, 2002488, C1

TEAM MEMBER

KULBHUSHAN, 2002496, C2

UNDER THE GUIDANCE OF

Ms. Jaspreet Kaur

Department of Computer Science & Engineering

Chandigarh Engineering College

Kharar-Banur Highway, Sector-112 Greater Mohali,

Punjab – 140307 (INDIA)

(Approved by AICTE, New Delhi and Affiliated to IKGPTU, Jalandhar)

DECEMBER 2020

CHANDIGARH ENGINEERING COLLEGE
CGC- LANDRAN, MOHALI

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the project report entitled "**SORTING VISUALIZER"** by **"KESHAV KUMAR"** in partial fulfillment of requirements for the award of degree of B.Tech. (CSE) submitted in the Department of CSE at Chandigarh Engineering College under PUNJAB TECHNICAL UNIVERSITY, JALANDHAR is an authentic record of my own work carried out during a period from **JANUARY 2021** to **JUNE 2021**_under the supervision of **Ms Jaspreet Kaur.**

KESHAV KUMAR
CSE-C1, 2002488

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge.
The B.Tech Viva –Voce Examination of **KESHAV KUMAR** has been held on__ and accepted.

Mrs. Jaspreet Kaur

ABSTRACT

Analysis and design of algorithms is a great challenge for both computer and information science students. Fear of programming, lack of interest, and the abstract nature of the programming concepts are the main cause of the high dropout and failure rates in introductory programming courses. With an aim to motivate and help students, many researchers have proposed different kinds of tools. Although it has been reported that some of these tools have a positive impact on gaining programming skills, the problem still remains mostly unsolved. This paper describes ViSA, a tool for the visualization of sorting algorithms. ViSA is an easy-to-setup and fully automatic visualization system with step-by-step explanations and comparison of sorting algorithms. Design principles and technical structure of the visualization system as well as its practical implications and educational benefits are presented and discussed.

ACKNOWLEDGEMENT

I would like to place on record my deep sense of gratitude to **Ms. Jaspreet Kaur** Dept. of Computer Science & Engineering, CEC-CGC, Landran for her generous guidance, help and useful suggestions.

I express my sincere gratitude to Dr. Sukhpreet Kaur, HoD in Department of CSE, CEC-CGC, Landran for her stimulating guidance, continuous encouragement.

I also wish to extend my thanks to **Ms. Jaspreet Kaur** for her insightful comments and constructive suggestions to improve the quality of this research work.

KESHAV KUMAR

CSE-C1, 2002488

CONTENTS

1. INTRODUCTION

1.1 Problem Definition

You are familiar with many types of sorting in your B. Tech CSE course in DSA(Data Structure & Algorithm) subject like Bubble sort, insertion sort , selection sort, quick sort and heap sort, etc. in which sorting means a Sorting Algorithm is used to rearrange a given array or list elements according to a comparison operator on the elements. The comparison operator is used to decide the new order of elements in the respective data structure. You can solve questions in theory, work on specific sorting and will get the answer but have you ever thought about how it works and how it's happening behind the process(mechanism of sorting). So we make a project to show or visualize you all the process of sorting that how sorting will work behind the theory part, through this project you also will get a deep understanding of such sorting algorithms & will get core idea of a particular algorithm, like Bubble sort, insertion sort , selection sort, quick sort and heap sort, etc.

1.2 Project Overview

We have learnt sorting algorithms like bubble sort, selection sort, insertion sort, quick sort, heap sort, etc. But often we fail to understand the core idea of a particular algorithm, maybe because we are unable to visualize how they work. So the most important thing to understand about these algorithms is visualization.
This project helps one to visualize a sorting algorithm.

Basically we design a UI (webpage) to show you the process of sorting with the help of HTML(Hyper -Text Mark Up Language), CSS (Cascading Style Sheet), and JavaScript.
We provide you four options to select :

(i) To select the size of the array.
(ii) To select the speed of the algorithm.
(iii) To generate a new random array instantly.
(iv) To select the sorting operation you want to apply or want to implement.

Sorting Operations we provide in our UI is :

(i) Bubble Sort
(ii) Selection Sort
(iii) Insertion Sort
(iv) Merge Sort
(v) Quick Sort
(vi) Heap Sort

Take a look at the working or on mechanism of providing sorting operations which will give you very deep, efficient and clear understanding.

Bubble Sort :
A bubble sort algorithm goes through a list of data a number of times, comparing two items that are side by side to see which is out of order. It will keep going through the list of data until all the data is sorted into order. Each time the algorithm goes through the list it is called a 'pass'.

Selection Sort :
Selection Sort is a type of sorting algorithm. ... This sorting algorithm is an in-place comparison-based algorithm in which the list is divided into two parts, **the sorted part at the left end and the unsorted part at the right**

end. Initially, the sorted part is empty and the unsorted part is the entire list.

Insertion Sort :
Insertion sort iterates, consuming one input element each repetition, and grows a sorted output list. At each iteration, insertion sort removes one element from the input data, finds the location it belongs within the sorted list, and inserts it there. It repeats until no input elements remain.

Merge Sort : Merge Sort is a divide and conquer algorithm. It works by recursively breaking down a problem into two or more sub-problems of the same or related type, until these become simple enough to be solved directly. The solutions to the sub-problems are then combined to give a solution to the original problem.

Quick Sort : Quicksort is an in-place sorting algorithm. ... Quicksort is a divide-and-conquer algorithm. It works by selecting a 'pivot' element from the array and partitioning the other elements into two sub-arrays, according to whether they are less than or greater than the pivot.

Heap Sort : Heapsort sorts an array by first converting the array into a heap so that it has the relational property described above. ... It then sorts the data in reverse by repeatedly placing the largest unsorted element into its correct place.

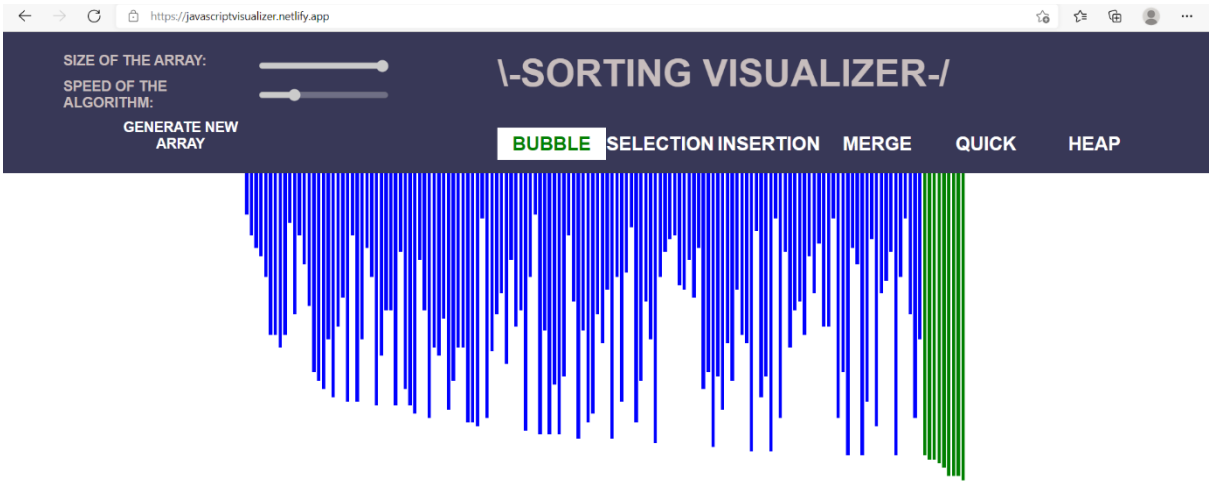
After done with selection of option and operations, sorting will start from left to right or right to left according to the rule of that particular sorting algorithm as you saw above. This sorting will work with respect to the speed of the algorithm you choose and for the certain time depending upon the speed and the size of array you choose. Means how many elements you choose in an array for sorting operation or you choose a random array or generate new random array size. Each element of the array is displayed as a bar.
Eventually we can say the size of the array is inversely proportional to the speed of the algorithm or vice versa.

Mathematically we can say :

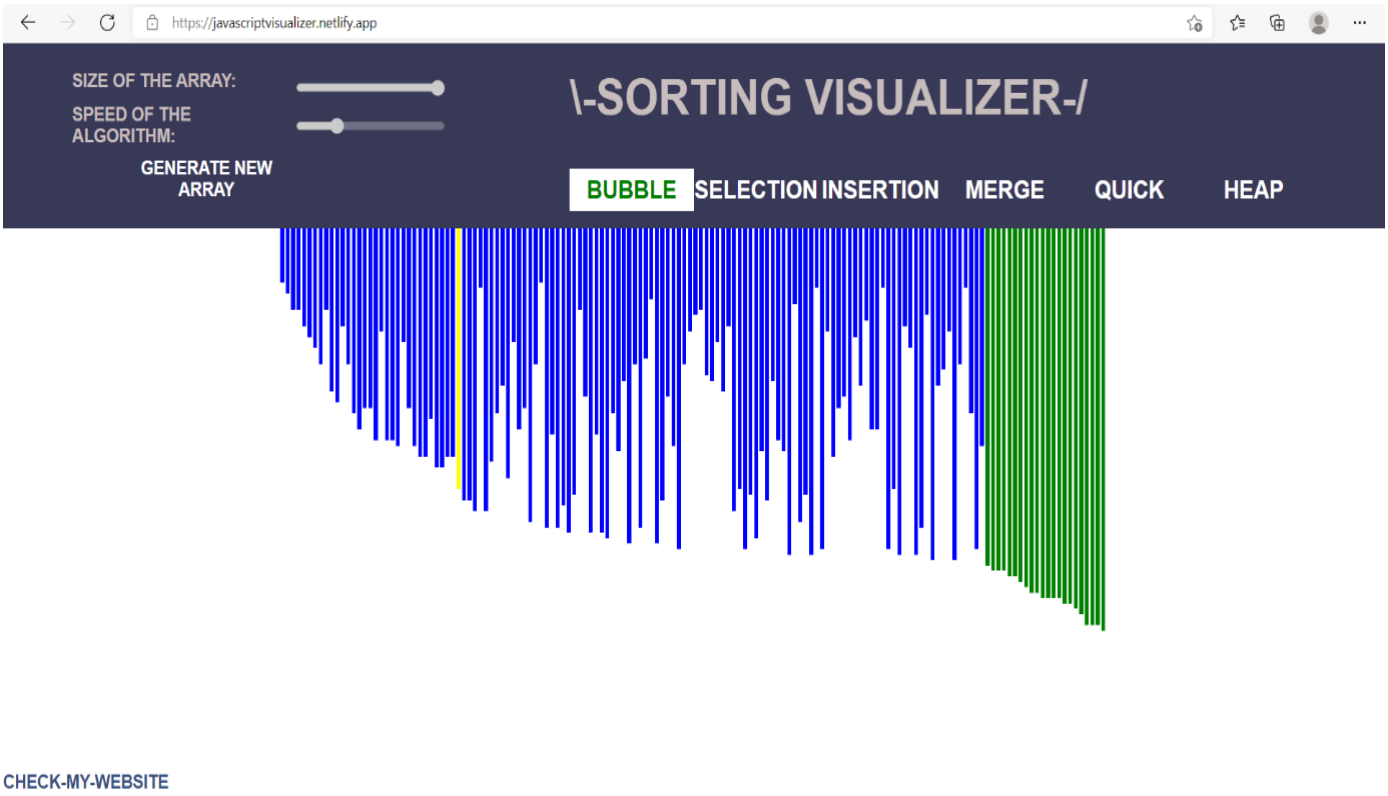
The size of the array $\propto$ 1/ speed of the algorithm.

As you can see in these two images which represents the sorting process and we select bubble sort.

In first image green color or right sided array bars are sorted with time.



Sorting process is going on and after 5 min it sorts a little more bars of array or array elements but takes too much time as discussed above.



Software/Hardware Specification :

Minimum Software/Hardware Configurations:

- Microsoft Windows Vista SP1/Windows 7 Professional
- Processor: 800MHz Intel Pentium III or equivalent
- Memory: 512 MB
- Disk space: 750 MB of free disk space
- Ubuntu 9.10:
- Processor: 800MHz Intel Pentium III or equivalent
- Memory: 512 MB
- Disk space: 650 MB of free disk space
- Macintosh OS X 10.7 Intel:
- Processor: Dual-Core Intel
- Memory: 2 GB
- Disk space: 650 MB of free disk space

Recommended Software/Hardware Configurations:

- Microsoft Windows 7 Professional/Windows 8/Windows 8.1:
- Processor: Intel Core i5 or equivalent
- Memory: 2 GB (32-bit), 4 GB (64-bit)
- Disk space: 1.5 GB of free disk space
- Ubuntu 15.04:
- Processor: Intel Core i5 or equivalent
- Memory: 2 GB (32-bit), 4 GB (64-bit)
- Disk space: 1.5 GB of free disk space
- OS X 10.10 Intel:
- Processor: Dual-Core Intel
- Memory: 4 GB
- Disk space: 1.5 GB of free disk space

1. LITERATURE SURVEY

2.1 Existing System

Many coding (technical) platforms have worked on it in JavaScript and Python language, like GeeksforGeeks, dev community, code projects. The code of this project is also available on code projects, Devcommunity and on Github in python language.
By knowing ResearchGate(website) three authors named Christopher Hundhausen, Sarah A. Douglas and John Staskowho worked for research(analysis) on it. They all are professors and researched in higher universities.
Even some of the youtube channels like codedrifter and Clément Mihailescu also work on it. This is a very easy way to learn and understand the working of sorting algorithms by this visualization project.
Useful websites describing the algorithms used:

- Bubble Sort (GeeksforGeeks)
- Selection Sort (GeeksforGeeks)
- Insertion Sort (GeeksforGeeks)
- Merge Sort (GeeksforGeeks)
- Quick Sort (GeeksforGeeks)
- HeapSort (GeeksforGeeks)

This project is based on HTML, CSS and JavaScript.

HTML was created by Tim Berners-Lee. The first version of HTML was written by Tim Berners-Lee in 1993. Since then, there have been many different versions of HTML. The most widely used version throughout the 2000's was HTML 4.01, which became an official standard in December 1999 HTML stands for Hyper Text Markup Language. It is the standard markup language for creating Web pages, describes the structure of a Web page, consists of a series of elements, elements tell the browser how to display the content and elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

CSS was first proposed by Håkon Wium Lie in 1994. At the time, Lie was working with Tim Berners-Lee at CERN. Several other style sheet languages for the web were proposed around the same time. Cascading Style

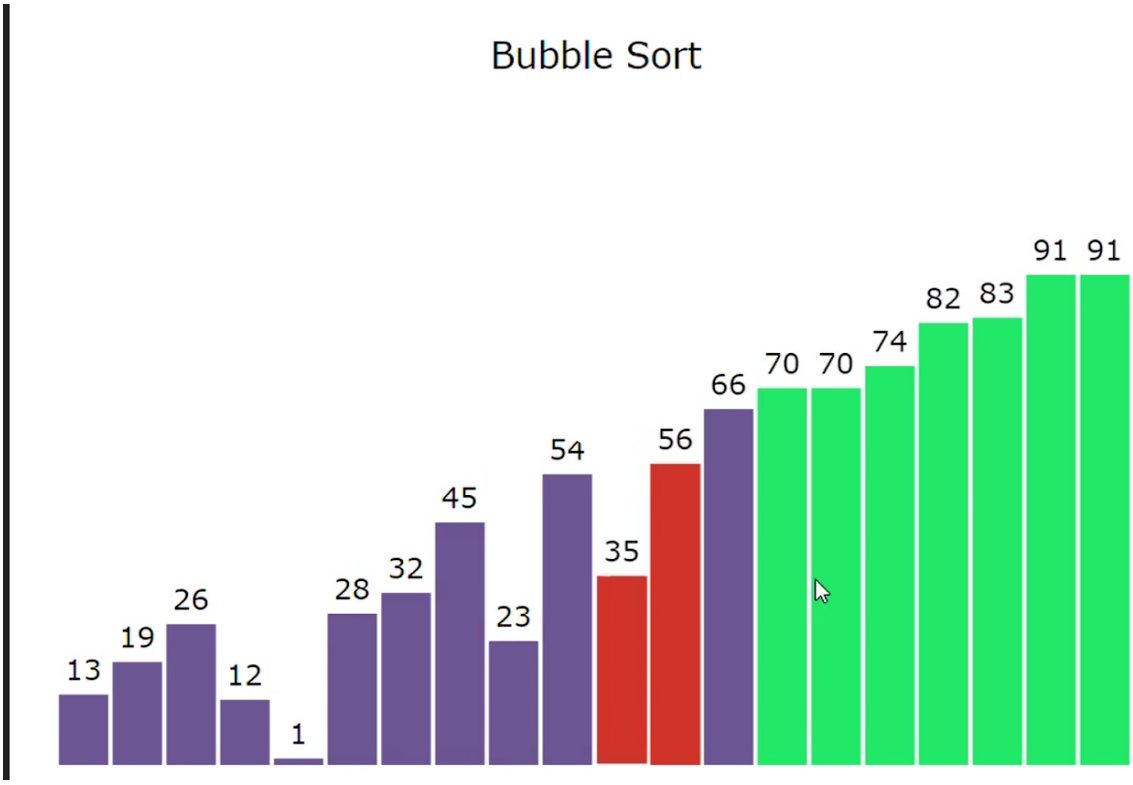
Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.
CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

Brendan Eich 1961 is an American computer programmer and technology executive. He created the JavaScript programming language and co-founded the Mozilla project, the Mozilla Foundation, and the Mozilla Corporation.
JavaScript is a text-based programming language used both on the client-side and server-side that allows you to make web pages interactive. Where HTML and CSS are languages that give structure and style to web pages.

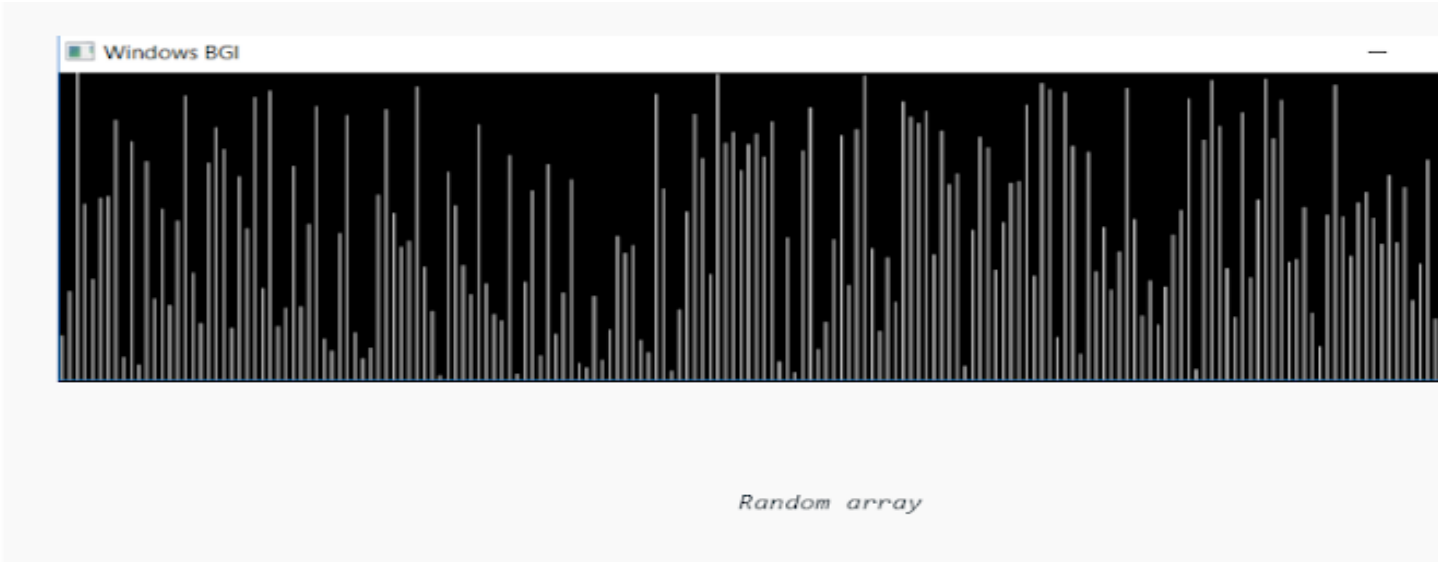
High-Level Approach: Creating the website's User Interface (UI) using HTML, CSS and enhancing it further using Bootstrap; without actually implementing any of the app's core features. Implementation of animations, effects and core functionalities (sorting algorithms) using JavaScript. This project is published to GitHub and also hosted.
GitHub- https://github.com/keshavkumar9431/sorting.visualizer
Hosted link- https://javascriptvisualizer.netlify.app/

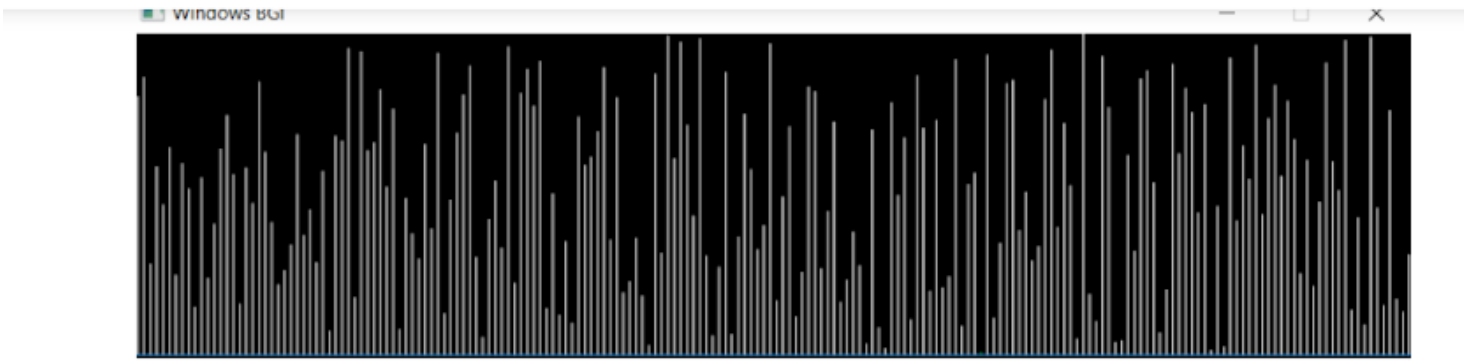
This is the already done example of sorting :

Bubble sort :

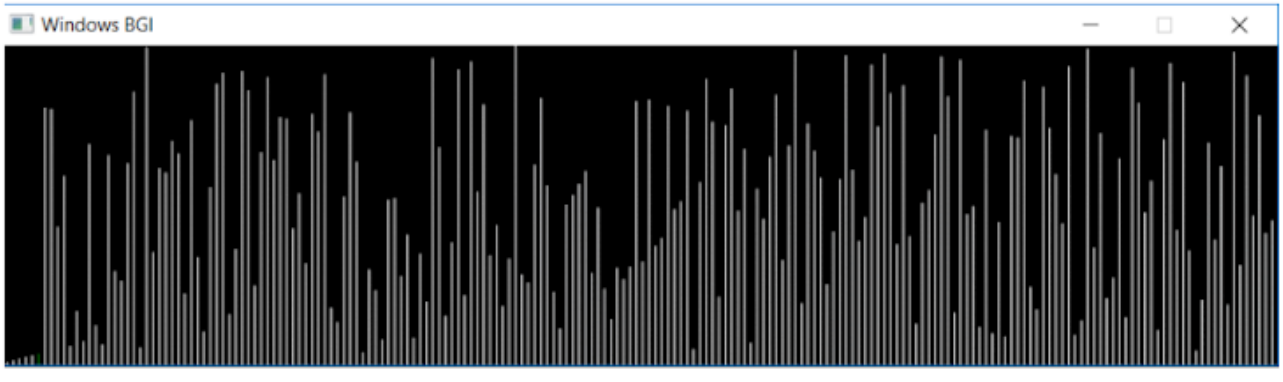


Selection sort :

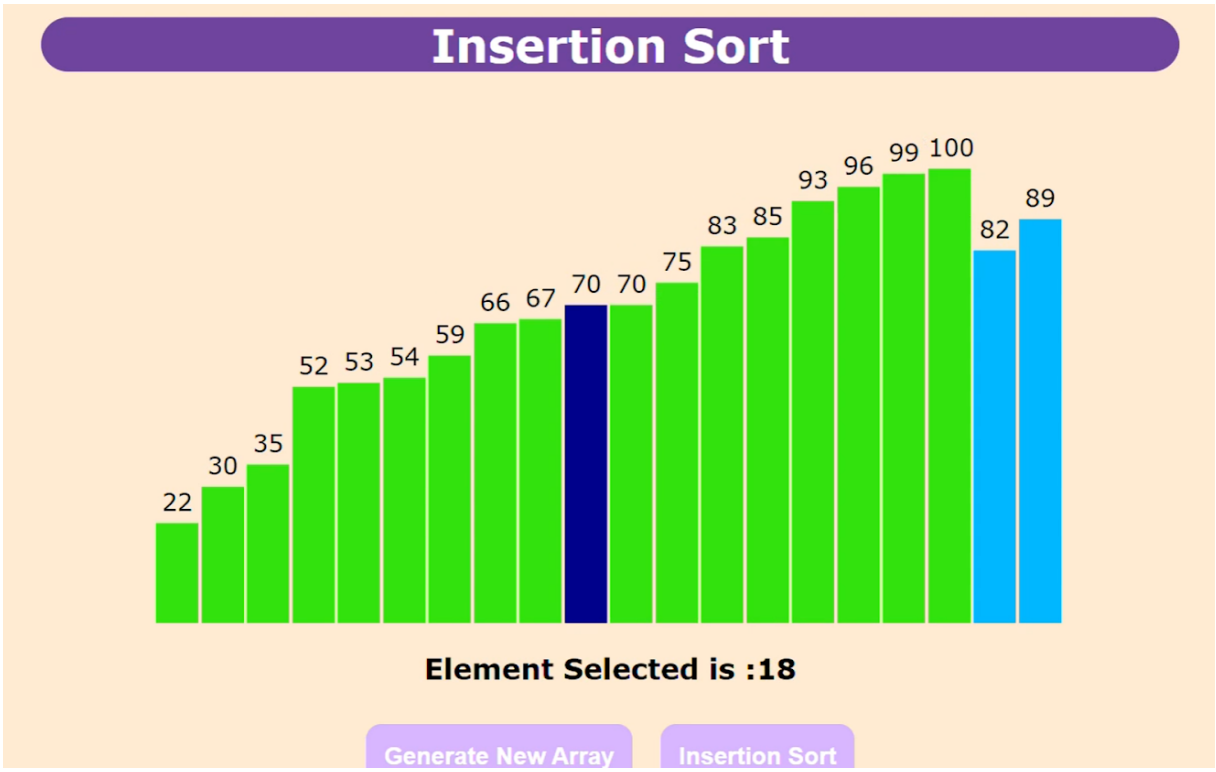




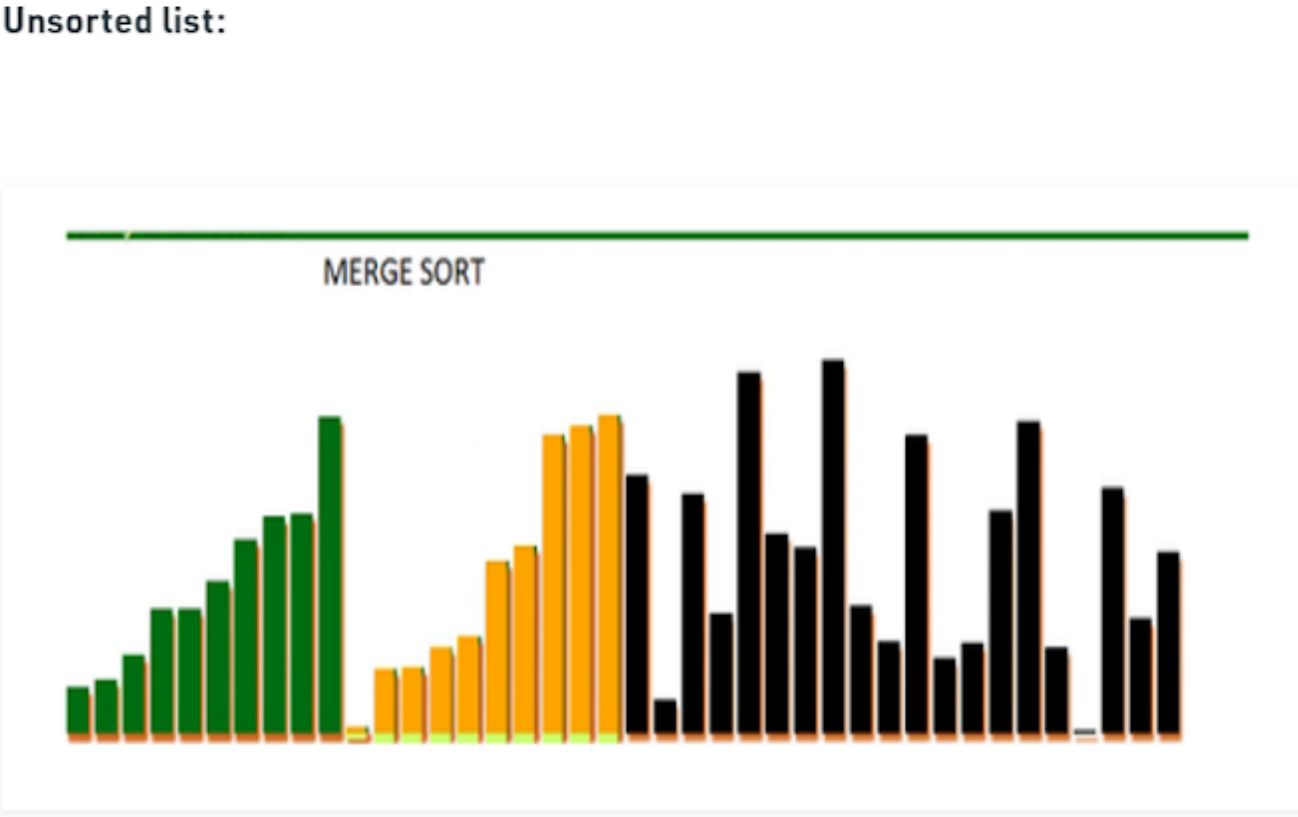
- **Output Visualization:**



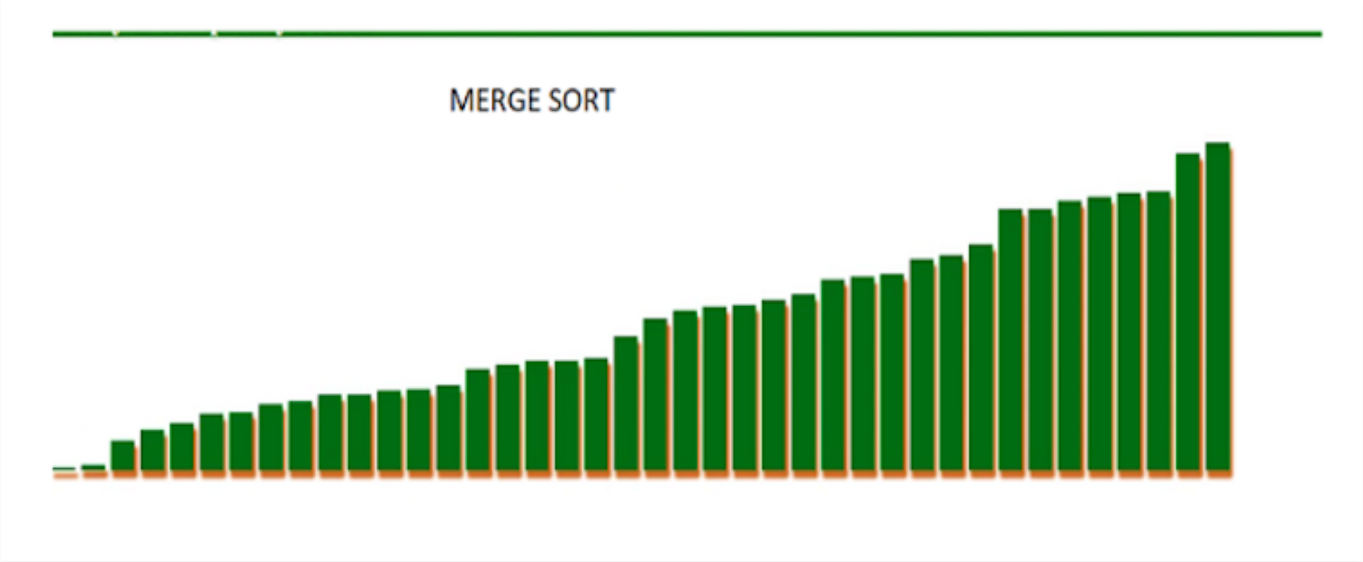
Insertion sort :



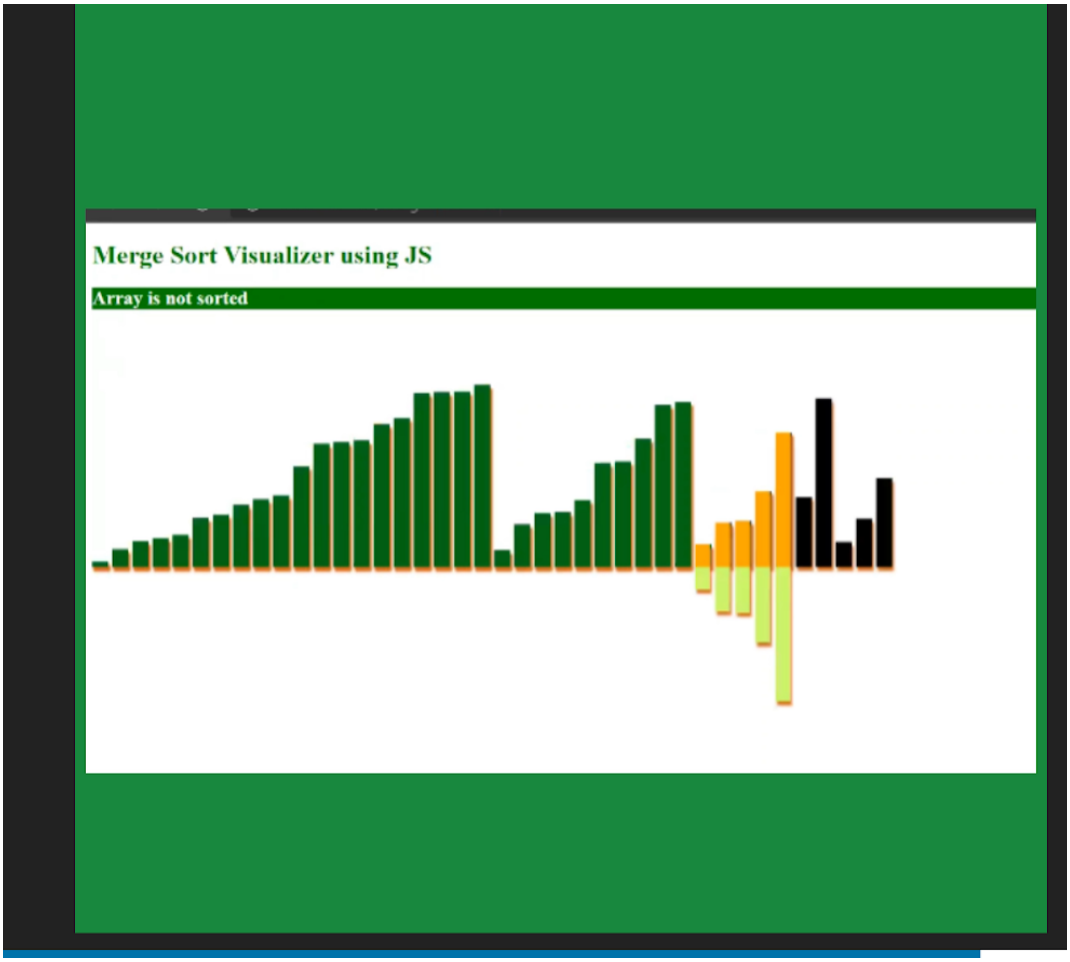
Merge sort :



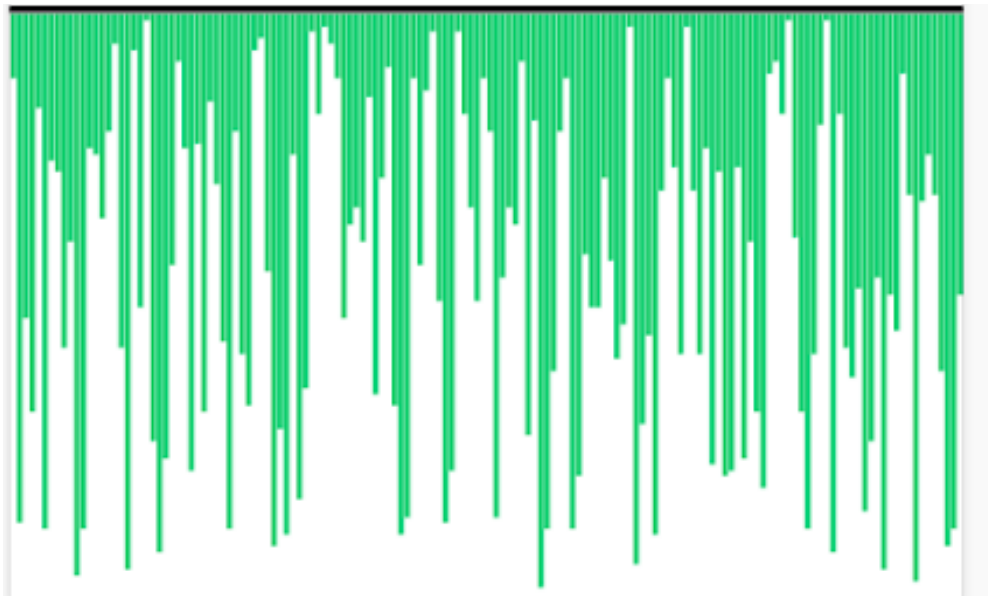
Sorted list:

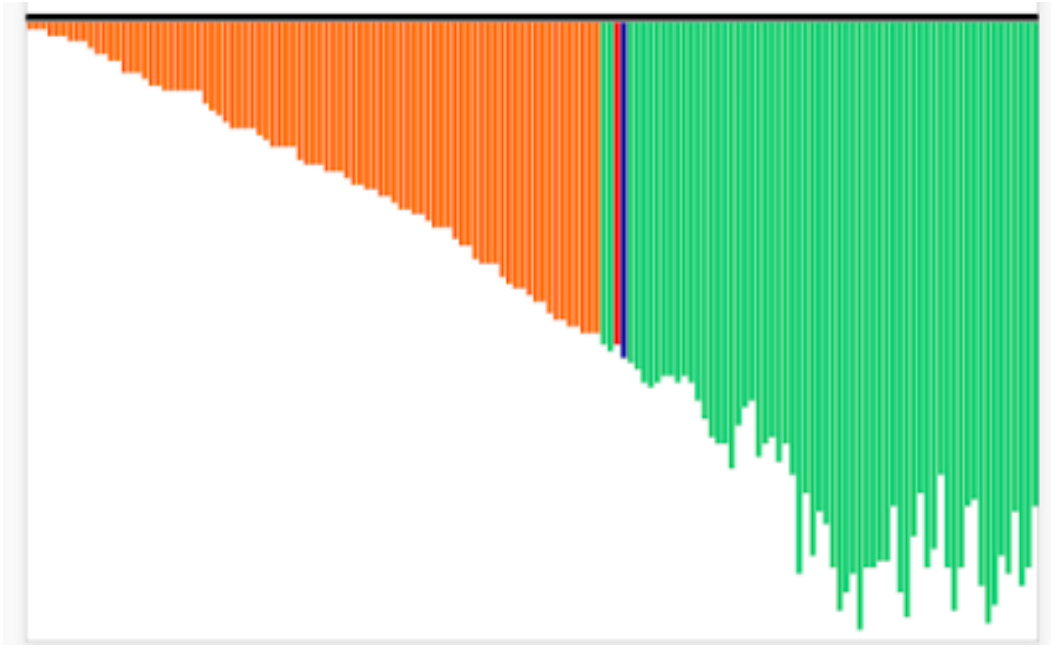


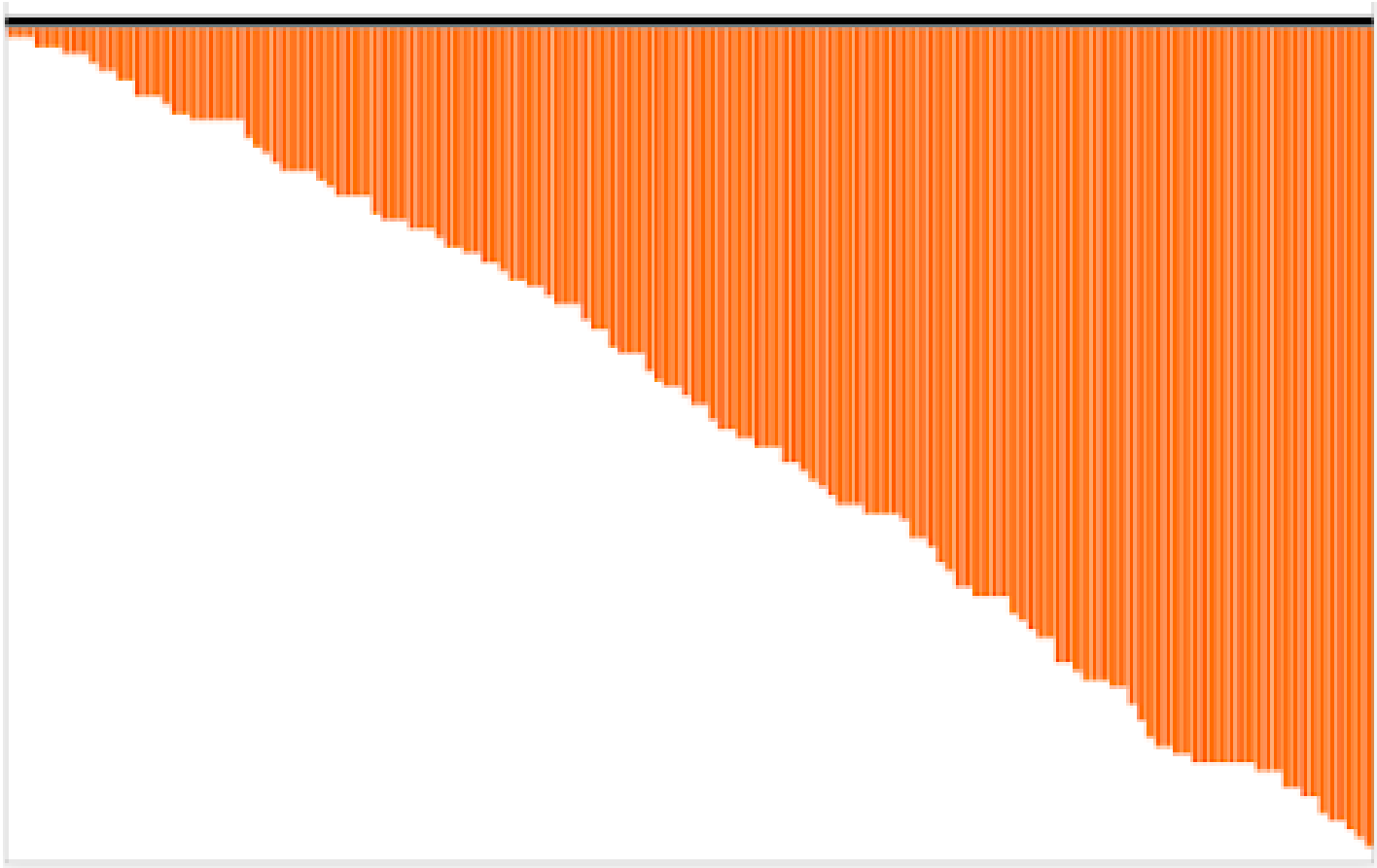
index.html: Below is the program to visualize the Merge Sort algorithm.

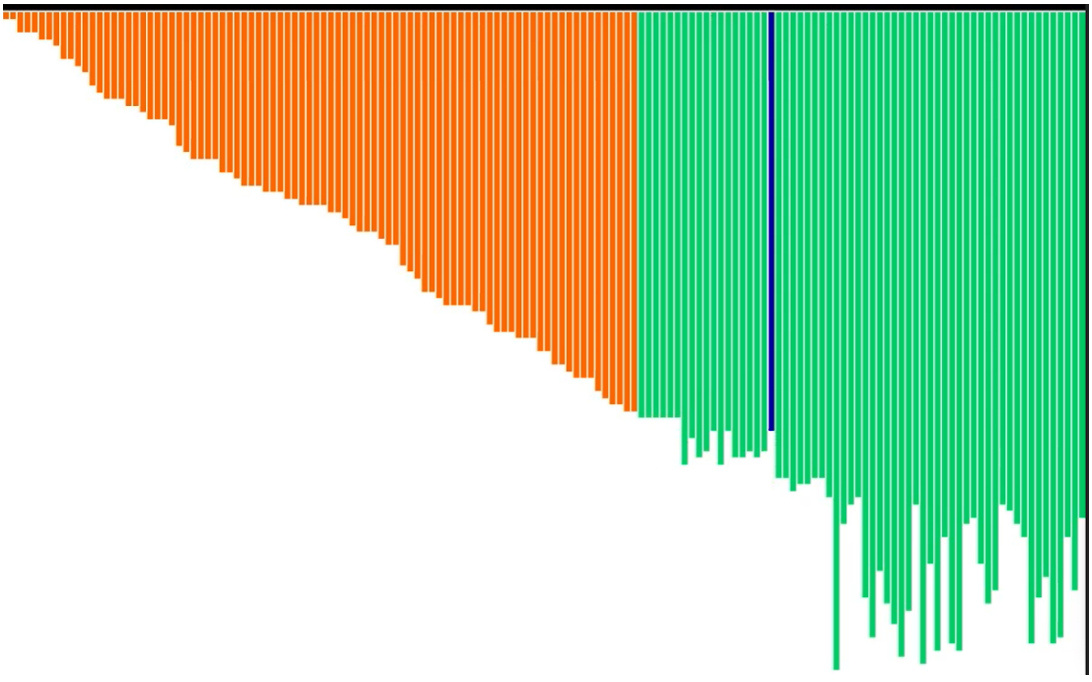


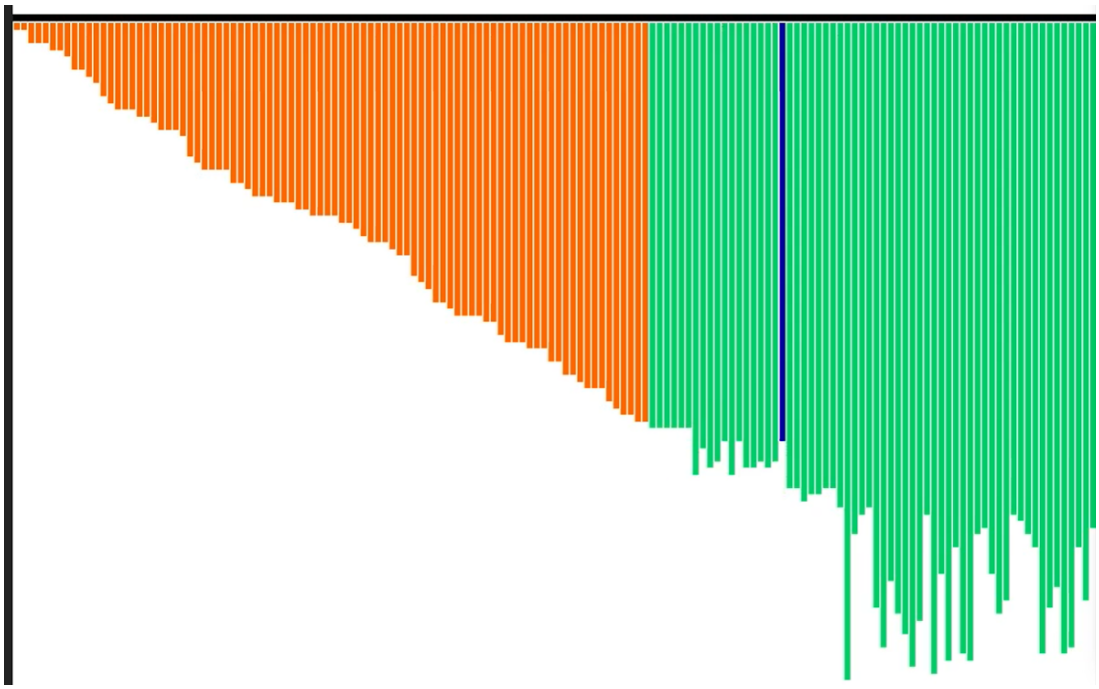
Quick sort :



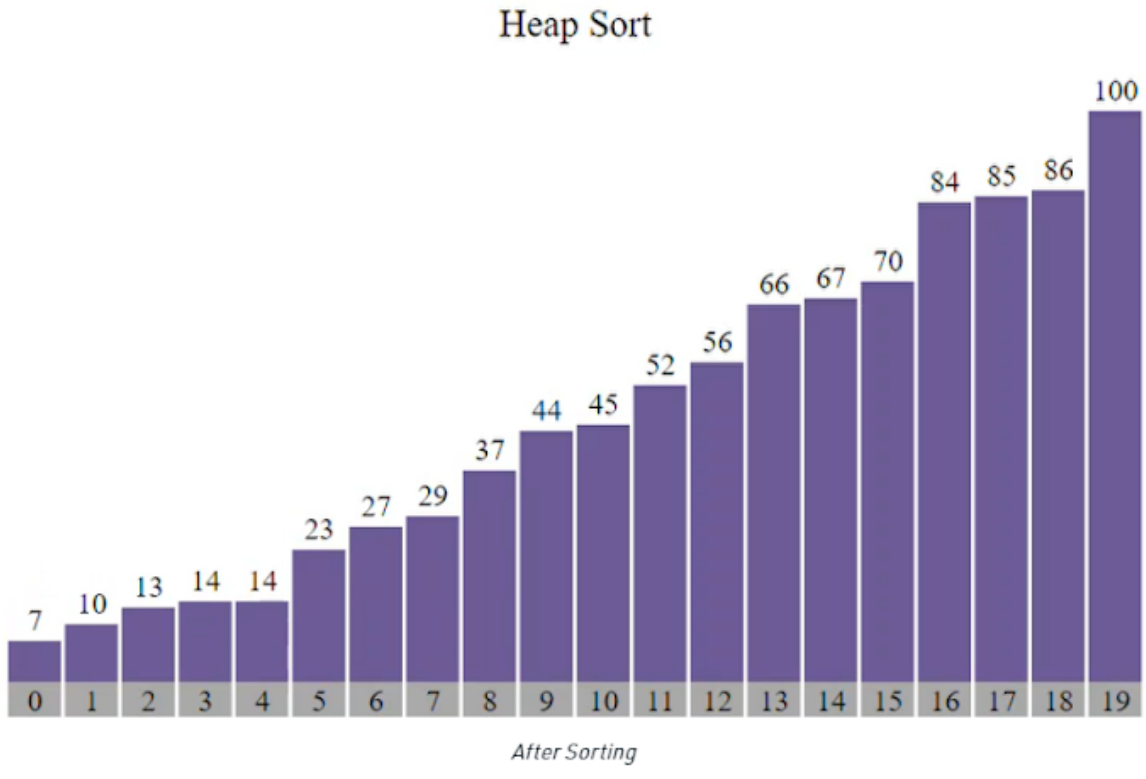




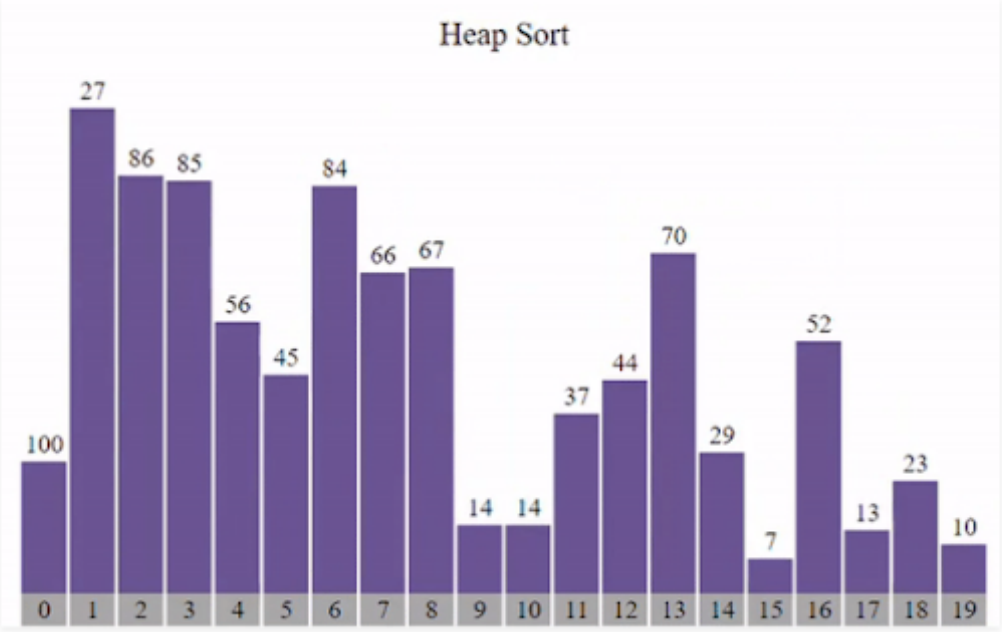




Heap Sort :


Heap Sort
7
10
13
14
14
23
27
29
37
44
45
52
56
66
67
70
84
85
86
100
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
After Sorting

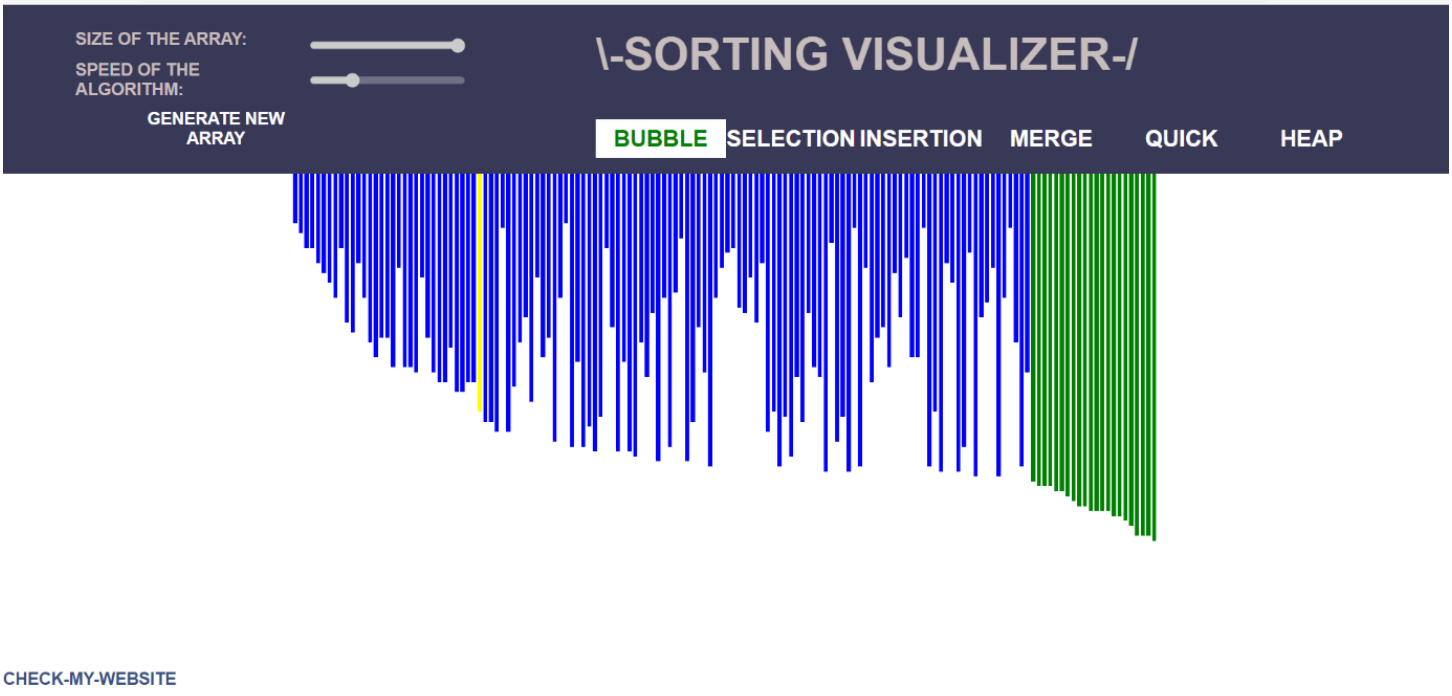
Output:


Heap Sort
100
27
86
85
56
45
84
66
67
14
14
37
44
70
29
7
52
13
23
10
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19

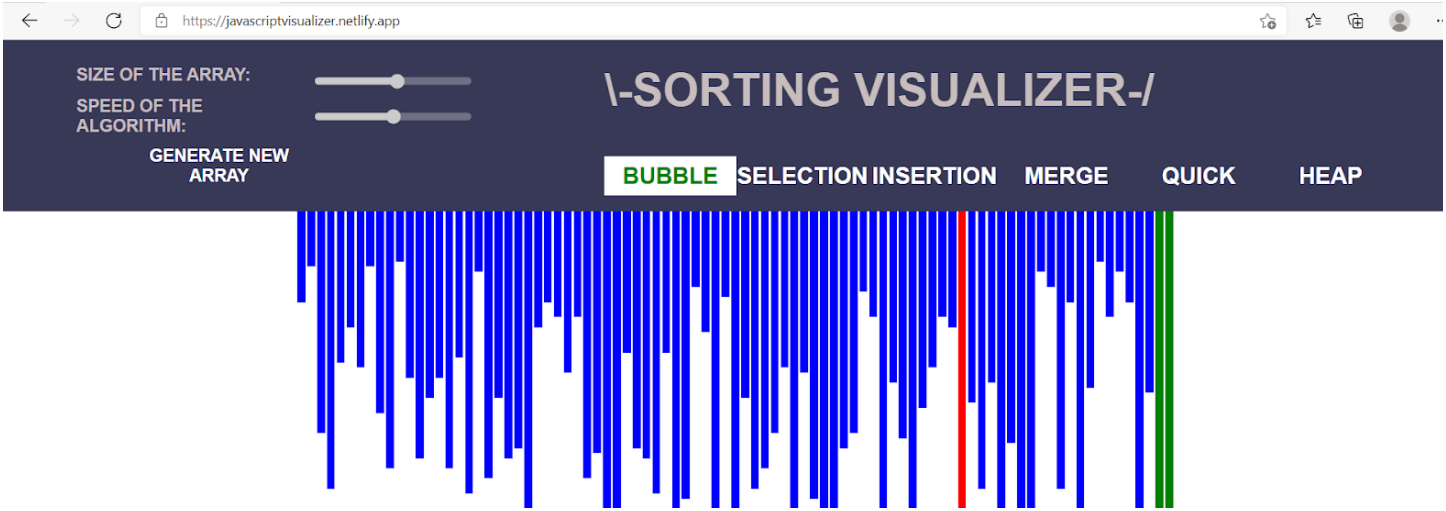
Methodology : In this section we show (describe) the working of specific sorting algorithms, like bubble sort, selection sort, insertion sort, merge sort, quick sort and heap sort etc. This images will provide you the core idea of sorting and give you a deep understanding of sorting algorithms visualization or sorting visualization.

We put here the every possible image of sorting, you can check below for specific sorting algorithm.

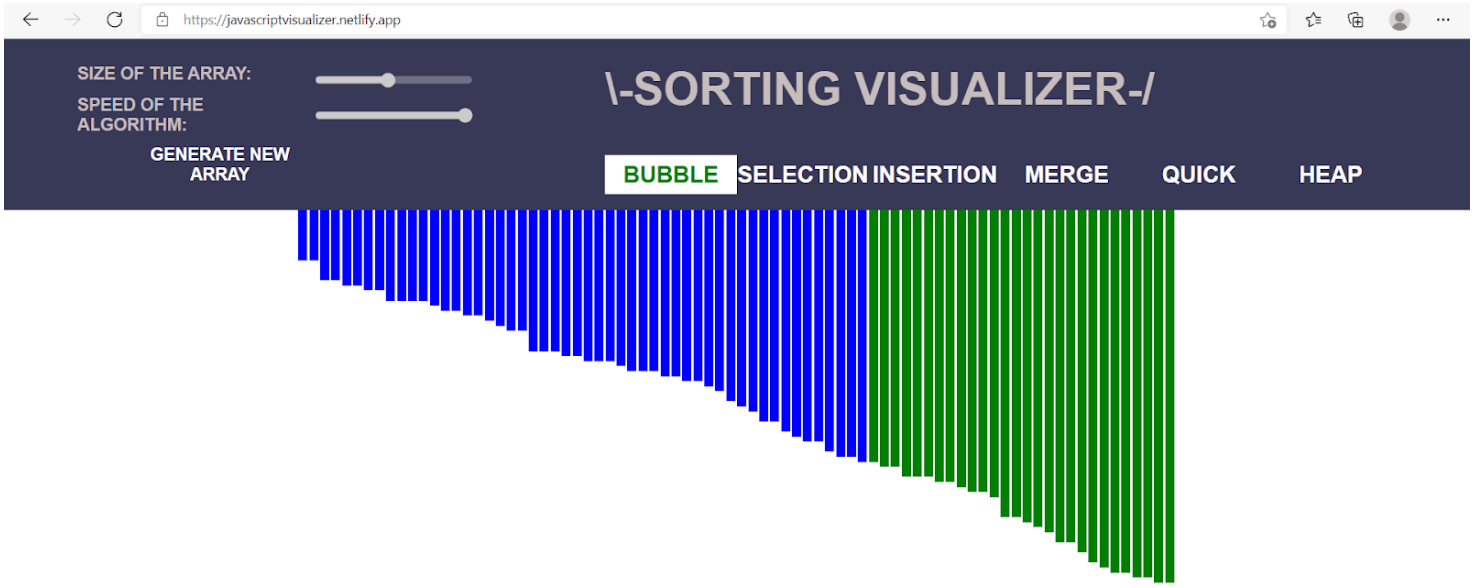
Bubble sort with low speed :



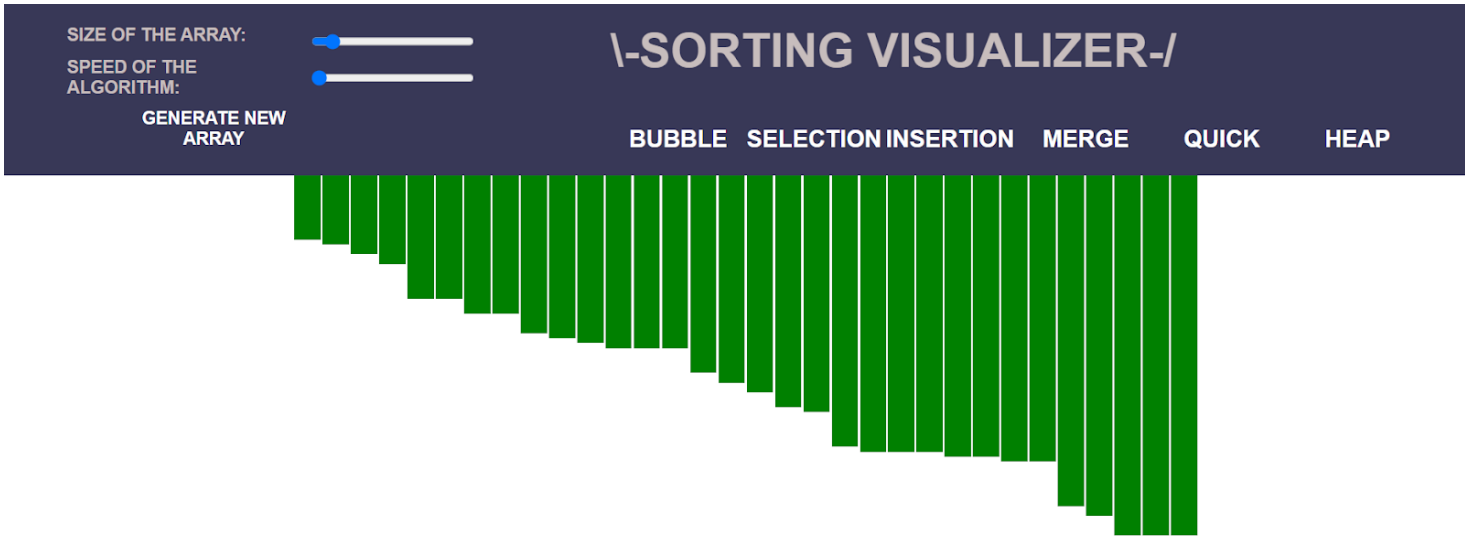
Bubble sort with moderate speed :



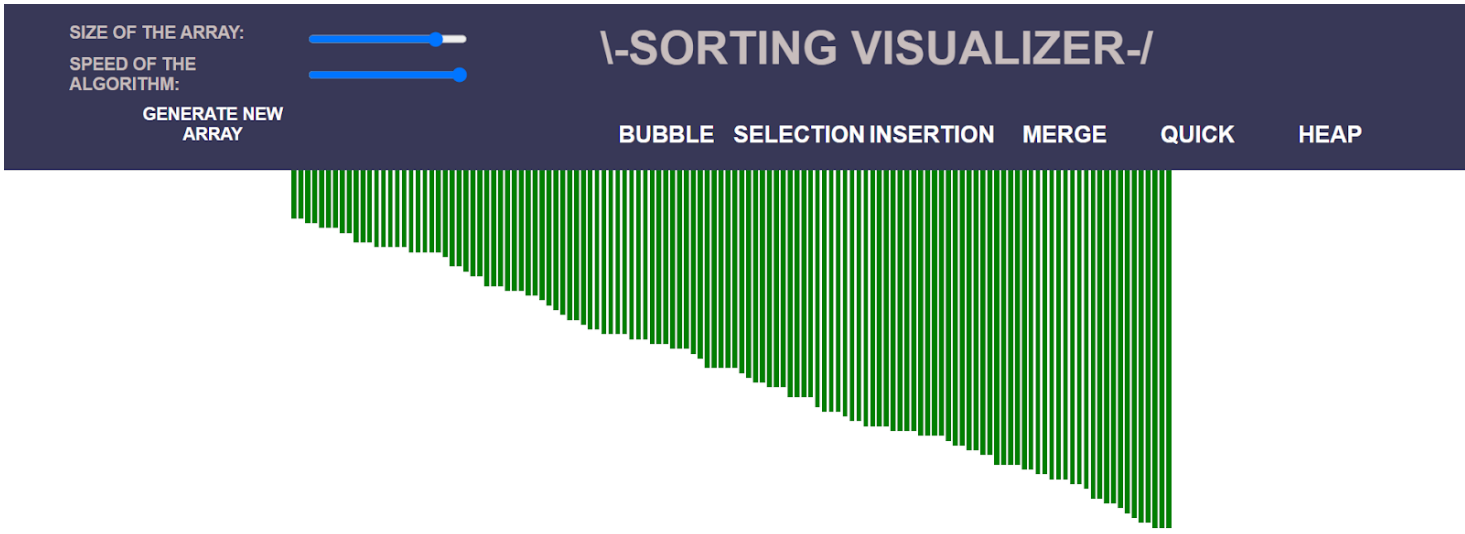
Bubble sort with fast speed :



Selection sort with low speed :

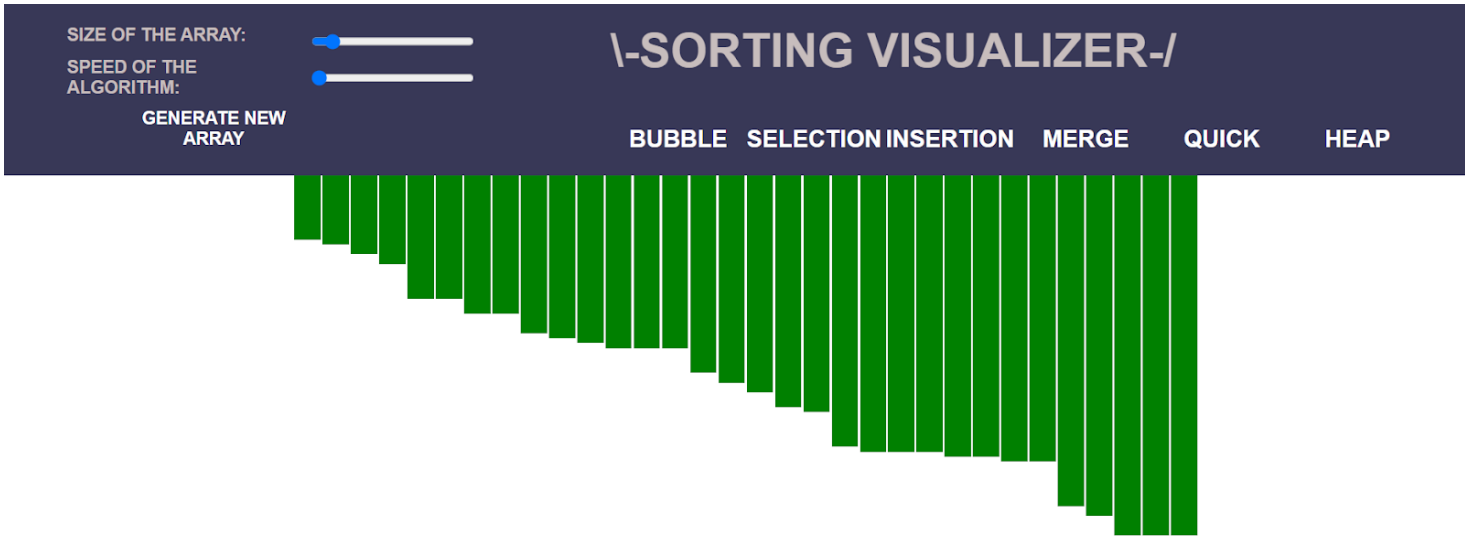


Selection sort with moderate speed :

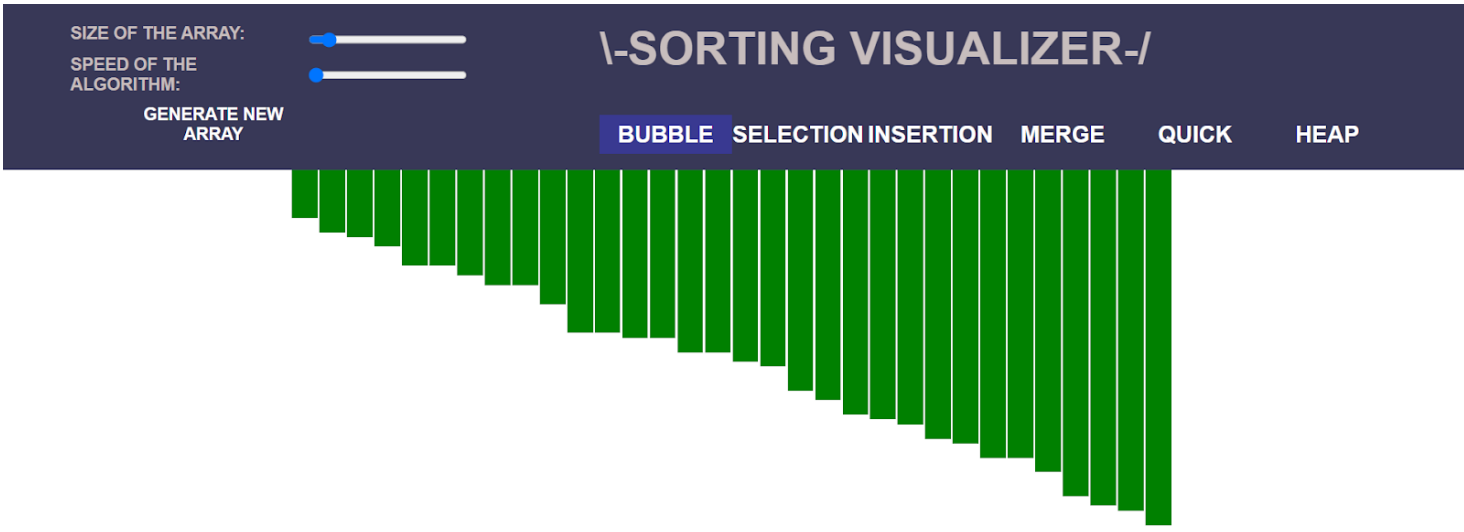


:HECK-MY-WEBSITE

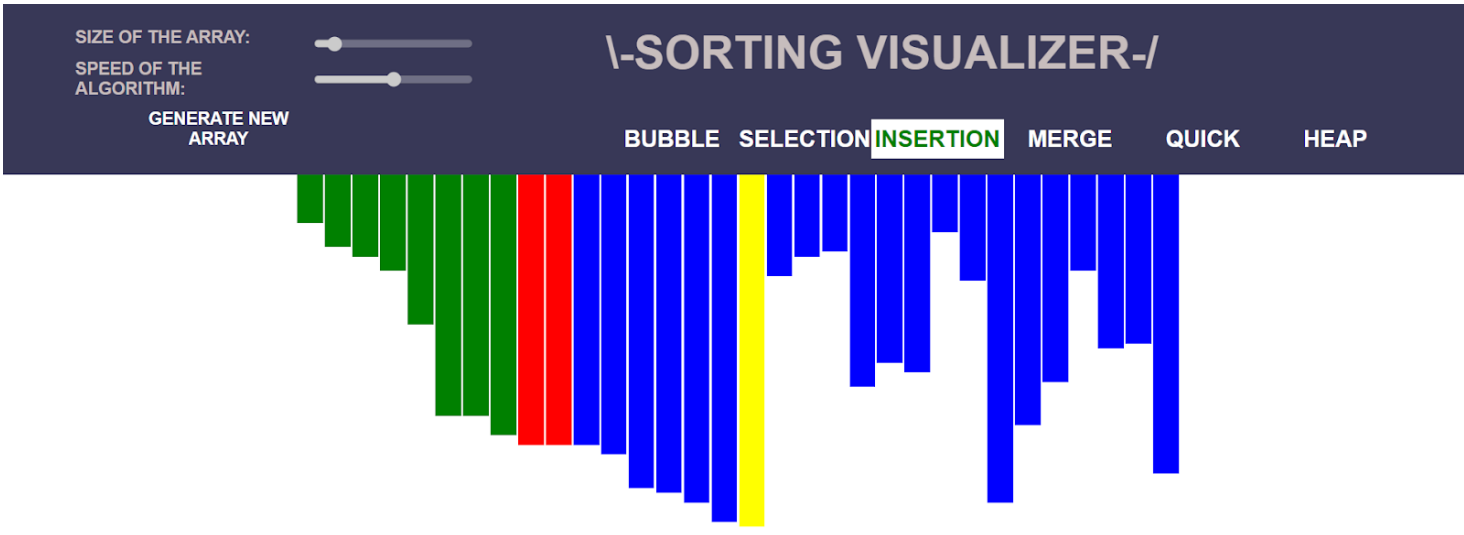
Selection sort with high speed :



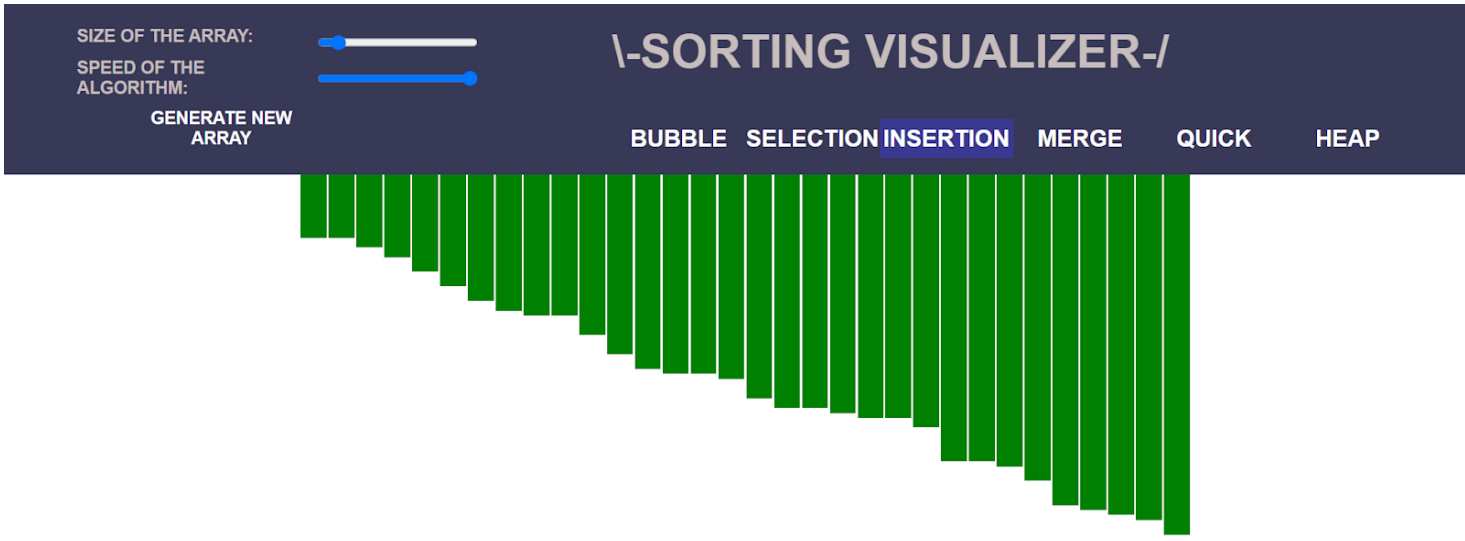
Insertion sort with low speed :



Insertion sort with moderate speed :

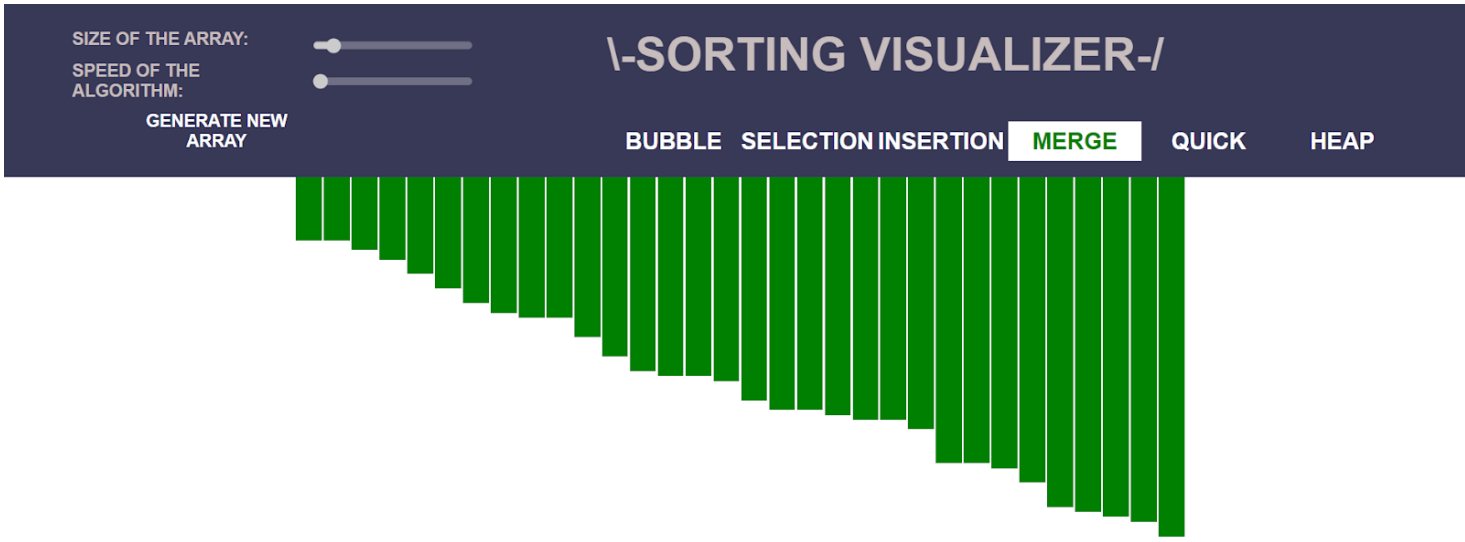


Insertion sort with high speed :



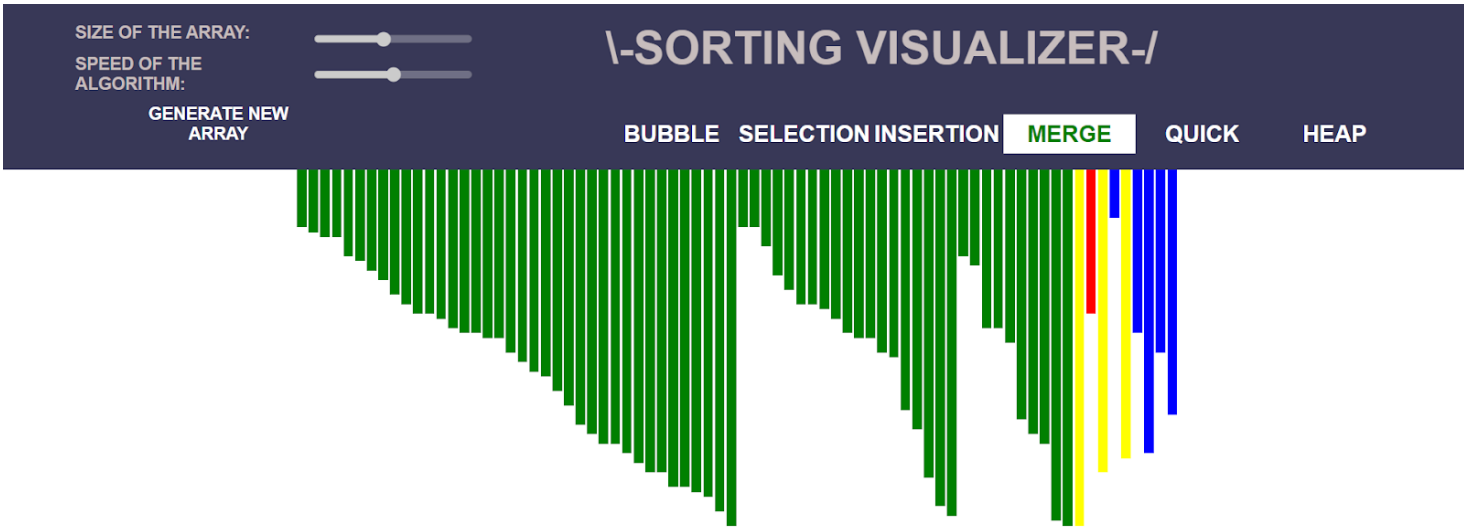
CHECK-MY-WEBSITE

Merge sort with low speed :



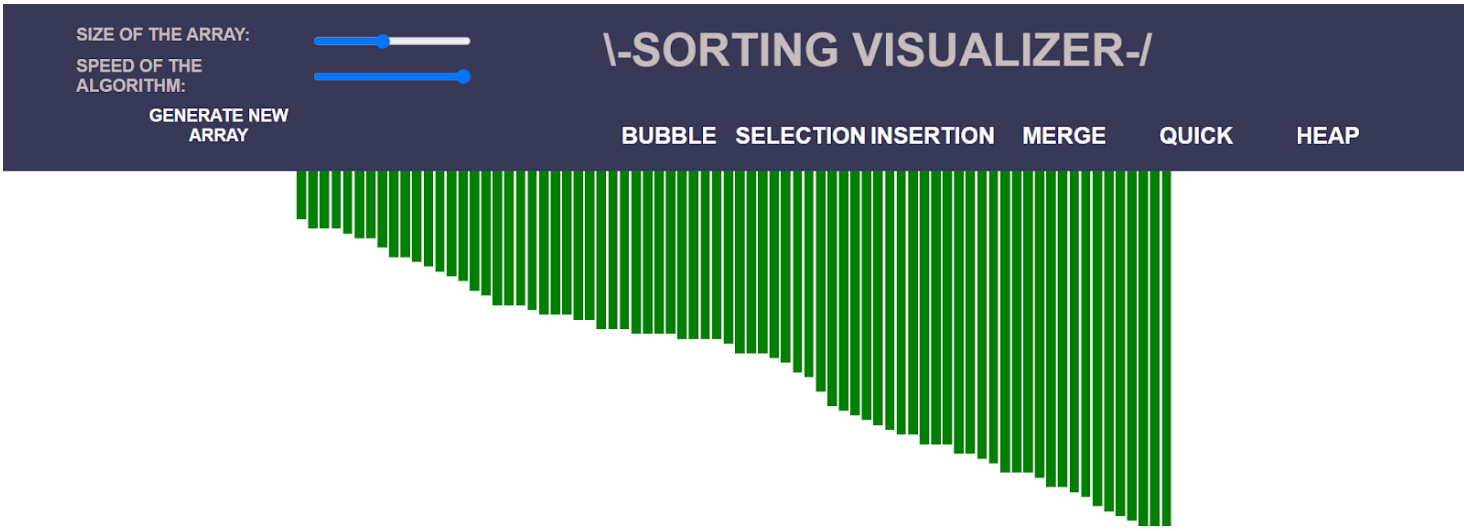
CHECK-MY-WEBSITE

Merge sort with moderate speed :



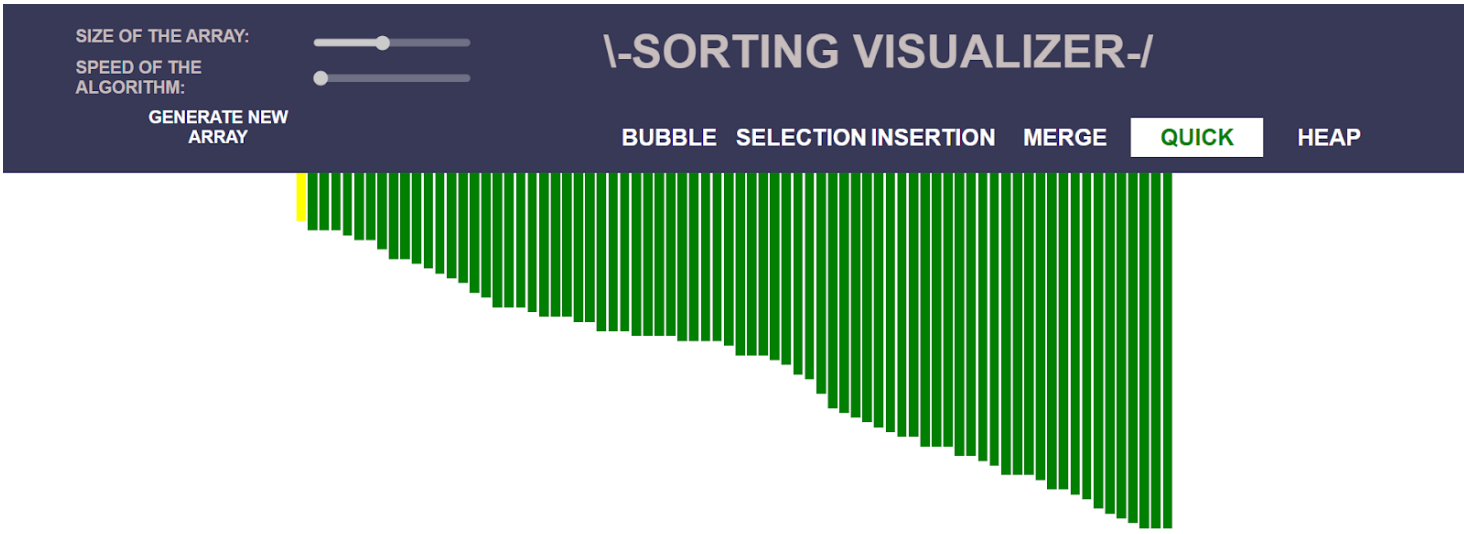
CHECK-MY-WEBSITE

Merge sort with high speed :

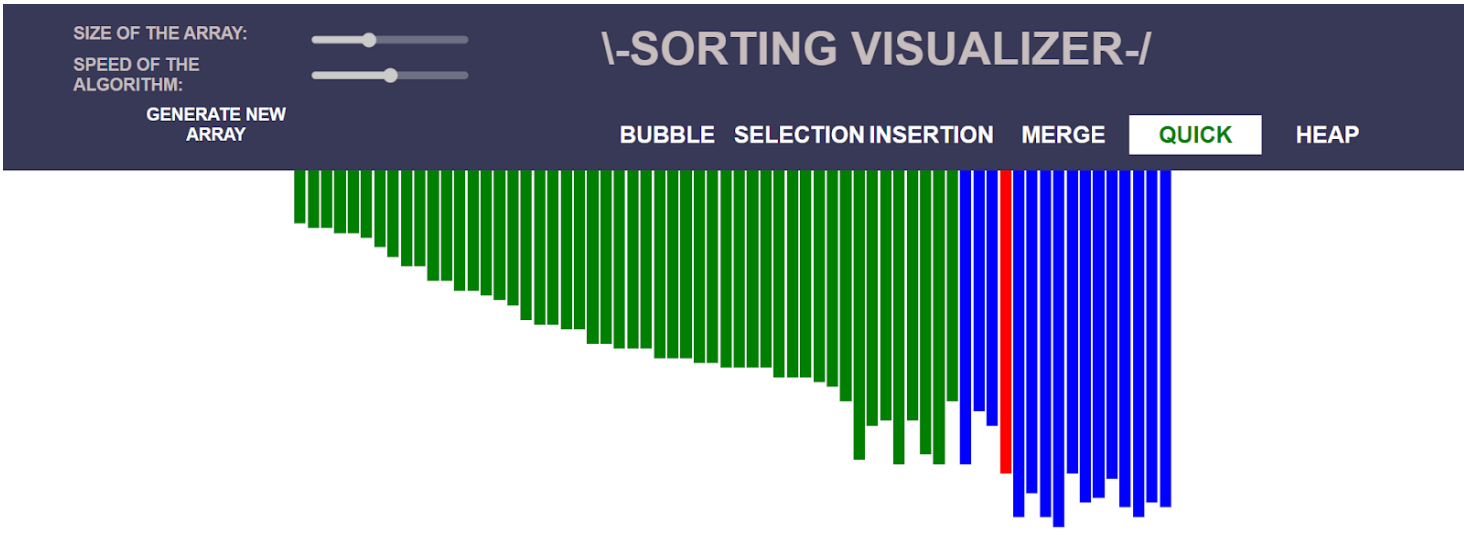


CHECK-MY-WEBSITE

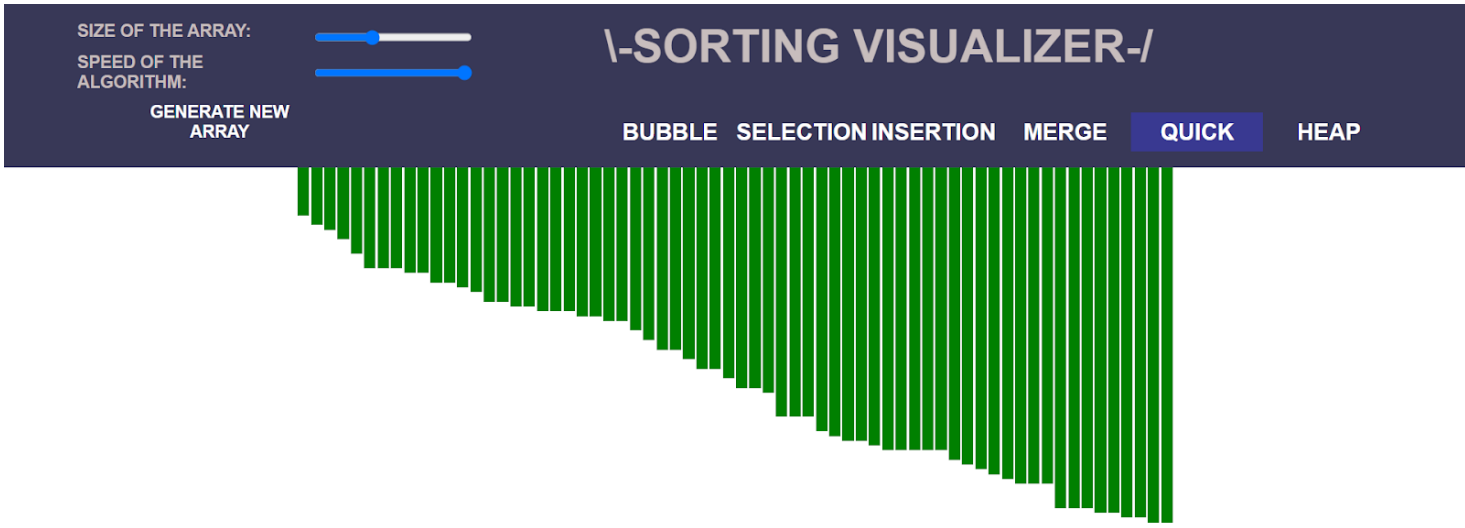
Quick sort with low speed :



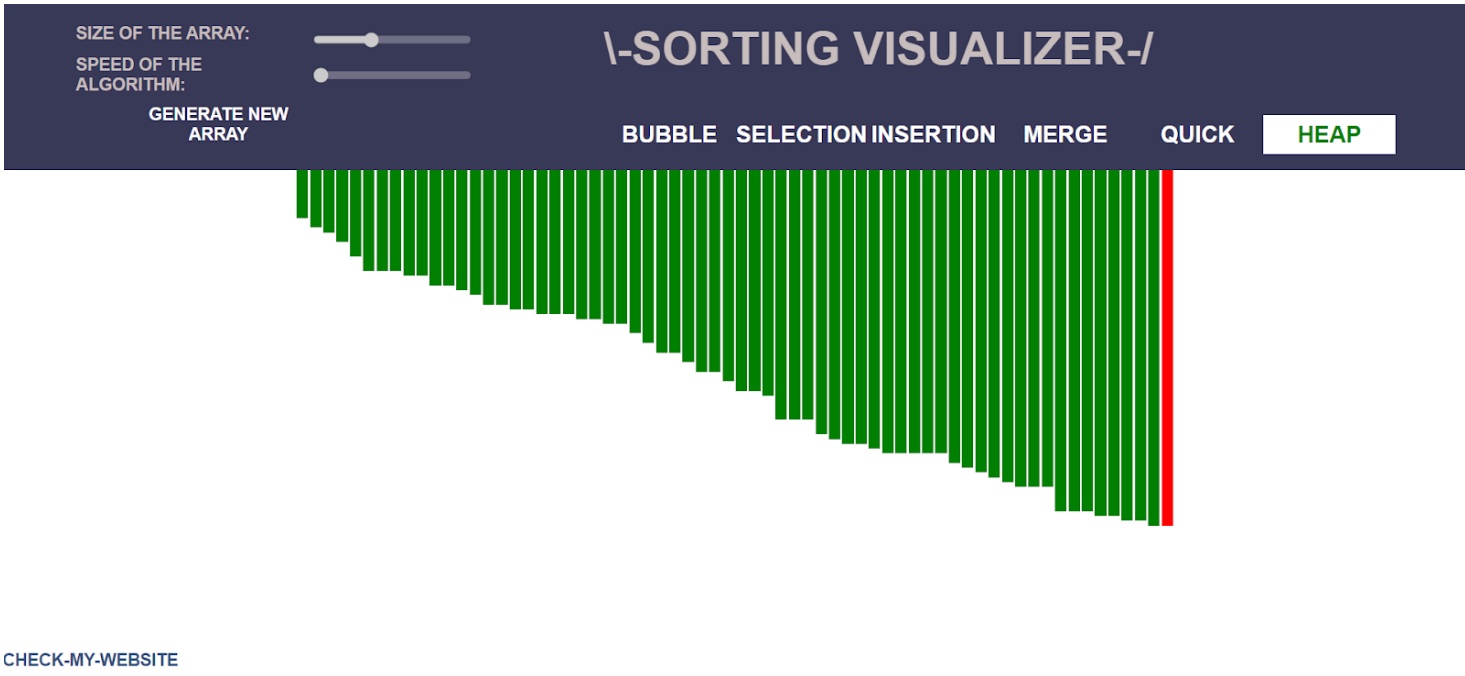
Quick sort with moderate speed :



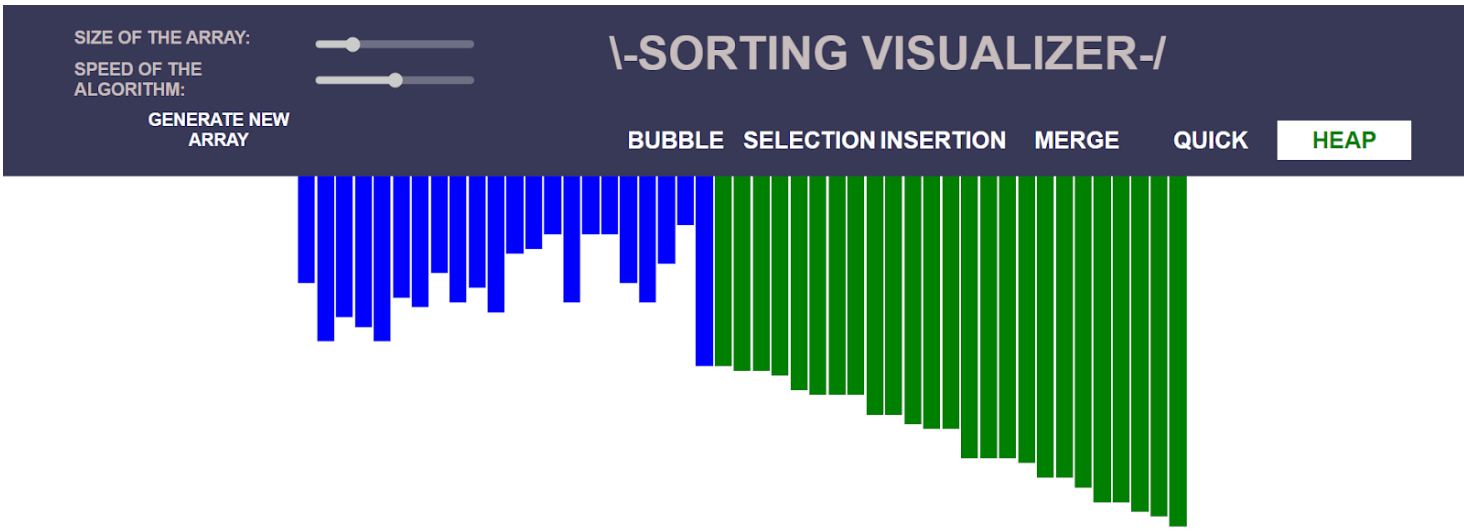
Quick sort with high speed :



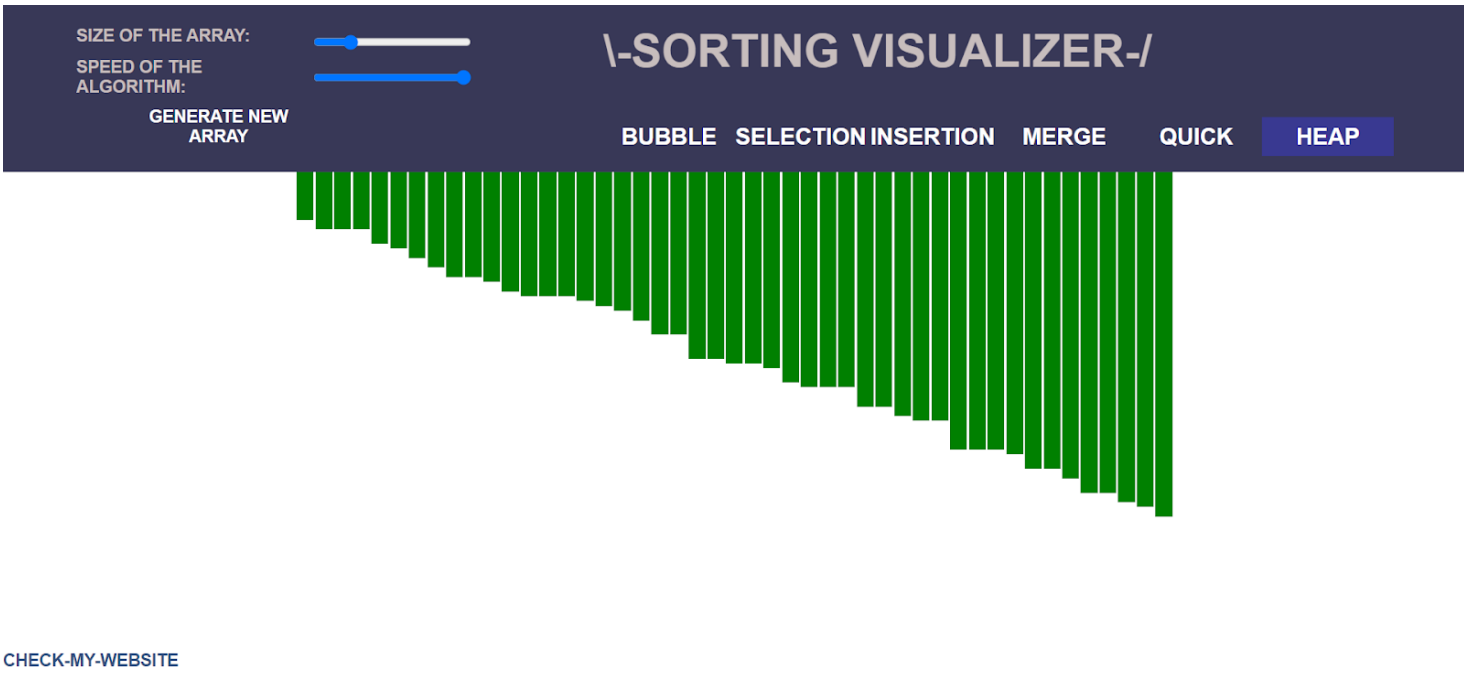
Heap sort with low speed :



Heap sort with moderate speed :



Heap sort with high speed :



CONCLUSION/RECOMMENDATION :

We have learnt sorting algorithms like bubble sort, selection sort, insertion sort, quick sort, heap sort, etc. But often we fail to understand the core idea of a particular algorithm, maybe because we are unable to visualize how they work. So the most important thing to understand about these algorithms is visualization. That's why we are making this project to let everyone understand how these algorithms work and through this project you also will get a deep understanding of such sorting algorithms. This project will guide you step by step to complete this project and at the end of this project you will have an immense grip on some core concepts of JavaScript as well. Adding this project on your resume will showcase your skills and add a great value to your profile. This project is a good start for beginners and a refresher for professionals who have dabbled in data structures and algorithms using JavaScript before and also web developers. This project is based on three languages as we design the UI of visualization which is HTML, CSS(Cascading Style Sheet), and JavaScript. These three languages are interconnected with each other without any one of them we can't make an awesome UI. HTML was created by Tim Berners-Lee. The first version of HTML was written by Tim Berners-Lee in 1993. Since then, there have been many different versions of HTML. The most widely used version throughout the 2000's was HTML 4.01, which became an official standard in December 1999 HTML stands for Hyper Text Markup Language. It is the standard markup language for creating Web pages, describes the structure of a Web page, consists of a series of elements, elements tell the browser how to display the content and elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc. CSS was first proposed by Håkon Wium Lie in 1994. At the time, Lie was working with Tim Berners-Lee at CERN. Several other style sheet languages for the web were proposed around the same time. Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. Brendan Eich 1961 is an American computer programmer and technology executive. He created the JavaScript programming language and co-founded the Mozilla project, the Mozilla Foundation, and the Mozilla Corporation. JavaScript is a text-based programming language used both on the client-side and server-side that allows you to make web pages interactive. Where HTML and CSS are languages that give structure and style to web pages.

REFERENCES :

Inspiration taken from Clément Mihailescu

(https://www.youtube.com/watch?v=pFXYym4Wbkc)

Useful links describing the algorithms used:

- Bubble Sort

(https://www.geeksforgeeks.org/sorting-algorithms-visualization-bubble-sort/)

- Selection Sort

(https://www.geeksforgeeks.org/selection-sort-visualizer-in-javascript/)

- Insertion Sort

(https://www.geeksforgeeks.org/insertion-sort-visualization-using-javascript/)

- Merge Sort

(https://www.geeksforgeeks.org/sorting-algorithm-visualization-merge-sort/)

- Quick Sort

(https://www.geeksforgeeks.org/quick-sortlomuto-partition-visualization-using-javascript/)

- Heap Sort

(https://www.geeksforgeeks.org/heap-sort-visualization-using-javascript/