

## Visualization of Sorting Algorithms with Animation in Web Application

A Muthukrishnan<sup>1</sup>, J Koti Vinayaka Naga Sai Dattu Pitchuka<sup>2</sup>, S Kamallesh<sup>3</sup>

<sup>1</sup>Faculty, Computer science and engineering Vel Tech Rangarajan & Dr.Sangunthala R&D Institute of Science and Technology, Chennai, India ,

<sup>2</sup>Student, Computer science and engineering Vel Tech Rangarajan & Dr.Sangunthala R&D Institute of Science and Technology, Chennai, India ,

<sup>3</sup>Faculty, IT Department, Velammal College of Engineering and Technology, Madurai, India ,

**Abstract**— Sorting visualizer project involves in the development of web based application that enables users to visualize various sorting algorithms. The application provides a graphical representation of the sorting process, which helps users to understand the working of the sorting algorithms better. This project is an excellent educational tool that helps students and developers to learn and understand various sorting algorithms such as Bubble Sort, Selection Sort, Insertion Sort, Merge Sort, Quick Sort, and many more. The sorting visualizer project is designed to provide a fun and interactive way to learn sorting algorithms. Users can select a sorting algorithm of their choice and see the algorithm in action. The application uses an array of random numbers that are then sorted using the selected sorting algorithm. As the sorting algorithm progresses, the application displays the array at each step, providing a graphical representation of the sorting process. This helps users to visualize the changes made to the array at each step of the sorting process.

**Keywords**—*Animated Graphs, Bubble Sort, Insertion Sort, Merge Sort, Quick Sort, Selection Sort, Sorting, Visualizer, Visual Studio, Web Application.*

### I. Introduction

There is a growing trend in web applications of displaying the algorithms behind the data. This can be done in a couple of ways: by using charts and graphs, or by writing code that creates the visualization itself. The “Sorting Visualizer” project is a web development project which is used to visualize the sorting algorithms and performs the operation of algorithms step by step as the algorithms work like Bubble Sort, Selection Sort, Merge Sort etc., The application allows both an interactive mode as well as an automated mode by collecting the data. When the users select the algorithm they want to view the application will run and display the information in real-time, such as the time taken to sort an array with a number of elements  $N$  using a particular algorithm. A drop down menu also allows the users to choose the data type of the array which need to be sorted. Generally, the sorting algorithms are very difficult to understand in the easiest way for the learners. Simple sorting algorithms are called so because they’re pretty easy to understand and implement. These algorithms are seldom used to sort large lists because of their quadratic time complexity. However, while sorting a small list, the runtime

doesn’t matter much, and we often use them for easier and quick implementation. In computer science, a sorting algorithm is an algorithm that puts elements of a list into an order. The most frequently used orders are numerical order and lexicographical order, and either ascending or descending. Efficient sorting is important for optimizing the efficiency of other algorithms (such as search and merge algorithms) that require input data to be in sorted lists. Sorting is also often useful for canonicalizing data and for producing human-readable output. A sorting algorithm will put items in a list into an order, such as alphabetical or numerical order. For example, a list of customer names could be sorted into alphabetical order by surname, or a list of people could be put into numerical order by age. Sorting a list of items can take a long time, especially if it is a large list.

### II. Ease Of Use

#### A. Aim of the project

The main objective of the Sorting visualizer project is to learn and visualise the sorting algorithms and how the algorithms works. We can see the steps of swapping or any divide and conquer part of these bars. Here are 7 different types of sorting algorithms is used to visualise. A visualization of

data is implemented as a bar graph, after which a data sorting and algorithm may be applied. The resulting animation is then performed either automatically or by the user, who then sets their own pace.

#### B. Project Domain

The project domain of sorting visualizer using HTML, CSS, and JavaScript is aimed at providing a visual representation of how different sorting algorithms work. The project is based on the idea of creating a user-friendly platform where individuals can learn and understand how different sorting algorithms work by simply seeing how data is sorted in real-time. The project domain encompasses different stages of development that include designing the user interface, coding, and testing the application

### III. Project Requirements

There are several existing methodologies for sorting visualizers, but the most common approach involves generating an array of random numbers and using a sorting algorithm to arrange the elements in ascending or descending order. The visualizer then displays the array at each stage of the sorting process, highlighting the elements being compared and swapped. Popular sorting algorithms used in visualizers include Bubble Sort, Insertion Sort, Selection Sort, Merge Sort, and Quick Sort. The visualizer can be implemented using various programming languages and frameworks, with some providing more advanced graphics and interactivity than others. Additionally, the visualizer can

be enhanced with features such as user input for array size and speed control.

#### A. Abbreviations and Acronyms

DSA - Data Structures and Algorithms  
HTML - Hyper Text Markup Language  
CSS - Cascading Style Sheets

JS - Java Script  
VS - Visual Studio

ROI - Return On Investment  
GUI - Graphical User Interface

#### B. System Specifications

- Processor: Intel Core i3 or AMD Ryzen 3 processor or higher.
- RAM: At least 4 GB of RAM
- Storage: At least 500 GB of hard disk space or 256 GB of SSD.

- Graphics: Integrated graphics card or dedicated graphics card with at least 1 GB of VRAM.
- Display: Minimum resolution of 1280 x 720 pixels or higher.
- Operating System: Windows 10, macOS or Linux.
- Web Browser: Chrome, Firefox, Safari, or Edge with WebGL support.
- Optional: A mouse and keyboard for better navigation and interaction.

#### C. Standards and Policies

VSCode Visual Studio Code (VS Code) is a widely used open-source code editor developed by Microsoft. It has several standards and policies that govern its usage and development. One of the key standards of VS Code is that it should be accessible to everyone regardless of their background or ability. It should also be fast, reliable, and easy to use. VS Code follows an open-source model, which means that its code is freely available for anyone to use, modify and distribute. Additionally, VS Code has a set of policies that govern how third-party extensions can interact with the editor, ensuring that they do not compromise its stability or security. Standard Used: ISO/IEC 27001.

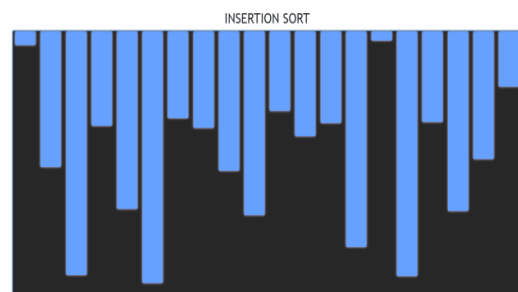
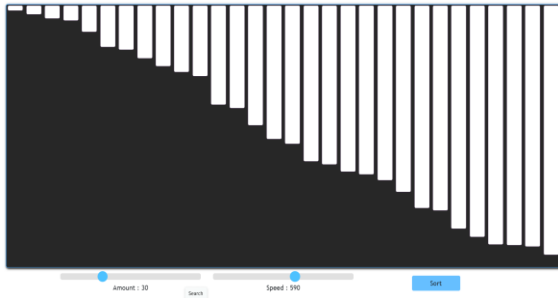


Figure 1: Sorting graphs of insertion sort on website

#### D. Module Description

- User Interface The frontend of sorting visualizer, and it involves designing the visual elements of the program using HTML and CSS. Creating an array of blocks or bars to represent the elements that need to be sorted and style them with CSS to give them the desired look and feel
- Input Validation This module involves validating the input provided by the user to ensure that it's valid and in the correct format. This is done using JavaScript and jQuery, which can check the data type of the input and ensure that it meets the required criteria



**Figure 2: Sorted bars after click on sort button**

- **Sorting Algorithms** This is the core of sorting visualizer and involves implementing different sorting algorithms such as bubble sort, insertion sort, and quick sort using JavaScript. The algorithms able to sort the elements in the array and update the UI to reflect the changes in real-time.
- **Control Panel** This is the module that provides the user with control over the sorting process. It includes features such as the ability to start, pause, and stop the sorting process, change the speed of the sorting, and select sorting algorithm. The control panel built using HTML, CSS, and JavaScript. Each of these modules developed and tested independently, and then integrated to form a complete sorting visualizer.

#### IV. Results And Discussions

Overall, sorting visualizers are a valuable resource for anyone who wants to learn about sorting algorithms or needs to sort large amounts of data. They provide a fun and interactive way to learn, and their applications are endless. With the help of sorting visualizers, users can improve their problem-solving skills and make better decisions based on data analysis.

##### A. Efficiency of Proposed System

The proposed system of sorting visualizer can greatly enhance the understanding and learning of sorting algorithms for beginners and experts alike. By providing a real-time visualization of the sorting process, users can easily see how the algorithm works and gain a deeper understanding of its efficiency and complexity. The ability to adjust the size and speed of the array also allows users to experiment with different scenarios and observe how the sorting algorithm performs under

different conditions. Furthermore, the inclusion of multiple sorting algorithm options, color-coding for visual emphasis, and user-friendly controls for customization would make the system highly versatile and user-friendly. The code traversal feature would also help users understand how the algorithm works under the hood, enabling them to learn the logic and implementation of the sorting algorithm. Overall, the proposed system of sorting visualizer has the potential to be a highly efficient and effective tool for learning and teaching sorting algorithms. Its interactive and visual nature can help users gain a deeper understanding of the sorting process and improve their skills in programming and data analysis.

##### B. Comparisons between proposed and existing system

**Existing system:** There are several existing methodologies for sorting visualizers, but the most common

approach is involving in generation of an array of random numbers and using a sorting algorithm to arrange the elements in ascending or descending order. The visualizer then displays the array at each stage of the sorting process, highlighting the elements being compared and swapped. Popular sorting algorithms used in visualizers include Bubble Sort, Insertion Sort, Selection Sort, Merge Sort, and Quick Sort. The visualizer can be implemented using various programming languages and 31 frameworks, with some providing more advanced graphics and interactivity than others. Additionally, the visualizer can be enhanced with features such as user input for array size and speed control.

**Proposed system:** The proposed system for sorting visualizer would involve implementing a web application that allows users to select a sorting algorithm and visualize the sorting process in real-time. The application would generate a random array of numbers and display it graphically on the screen, with options for adjusting the size and speed of the array. As the sorting algorithm runs, the application would highlight the elements being compared and swapped, providing a step-by-step visualization of the sorting process. Once the sorting is complete, the application would display the sorted array along with the number of swaps and comparisons made during the process. The proposed system could also include additional

features such as multiple sorting algorithm options, color-coding for visual emphasis, user-friendly controls for customization, also code traverses parallelly along with the sorting bars.

### Conclusion

In conclusion, sorting visualizers are a great way to learn about different sorting algorithms and how they work. They provide a visual representation of the sorting process, which makes easier for learners to understand the concept. Visualizing the sorting process can help learners identify the strengths and weaknesses of each algorithm and how they perform under different conditions. Furthermore, sorting visualizers can be used as a tool for problem-solving, data analysis, and decision making. They allow users to test and compare different sorting algorithms and choose the most efficient one based on their needs. Visualizers also help users identify patterns in data and make predictions about future outcomes. Overall, sorting visualizers are a valuable resource for anyone who wants to learn about sorting algorithms or needs to sort large amounts of data. They provide a fun and interactive way to learn, and their applications are endless. With the help of sorting visualizers, users can improve their problem-solving skills and make better decisions based on data analysis.

### Acknowledgment

Once the sorting is complete, the application would display the sorted array along with the number of swaps and comparisons made during the process. The proposed system could also include additional features such as multiple sorting algorithm options, color-coding for visual emphasis, user-friendly controls for customization, also code traverses along with the sorting bars.



```
CODE
C C++ Java Python
Bubble Selection Insertion Quick
// C program for implementation of Selection sort
#include <stdio.h>
void swap(int *xp, int *yp)
{
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}
void selectionSort(int arr[], int n)
{
    for (int i = 0; i < n-1; i++)
    {
        // One by one move boundary of unsorted subarray
        // Find the minimum element in unsorted array
```

Figure 3: Code presentation of a sort

### References

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] was the first ...”

- [1] Abawajy, J. H., Kelarev, A. (2018). Visualization of sorting algorithms: A survey. Journal of Visualization, 21(4), 699-717.
- [2] Ahmad, M. S., Arshad, S. (2018). Sorting algorithms visualization using Canvas and Javascript. In 2018 15th International Bhurban Journal on Applied Sciences and Technology (IBCAST) (pp. 332- 338).
- [3] Bowers, J., Rieger, D., Beall, C. (2017). Visualization techniques for sorting algorithms: A survey. IEEE Transactions on Visualization and Computer Graphics, 23(1), 341-350.
- [4] Chong, M. W., Tan, H. K. (2019). Interactive virtualization of sorting algorithms using virtual reality. In 2019 IEEE 9th International Journal on Cybernetics and Intelligent Systems (CIS) and IEEE Journal on Robotics, Automation and Mechatronics (RAM) (pp. 384- 389)
- [5] Garcia, D. R., Jia, X. (2018). Sorting algorithm visualization using augmented reality. In 2018 IEEE International journal on Teaching, Assessment, and Learning for Engineering (TALE) (pp. 289-293)
- [6] Li, Y., Li, D., Li, J. (2019). Visualization of sorting algorithms based on HTML5 canvas. In 2019 5th International Journal on Control, Automation and Robotics (ICCAR) (pp. 530-535)
- [7] Manasseh, M., Tahir, N. A. (2018). Algorithms for sorting: A literature review. Journal of Computer Science and Technology, 18(1), 1-22. Shu, R., Wang, X. (2019). A comparative study of sorting algorithms using visualization techniques. Journal of Software Engineering and Applications, 12(1), 29