IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 11, Issue 6, June 2024

DOI: 10.17148/IARJSET.2024.11634

Sortify: Visualising Algorithms

Nandini G R[1], Chandana R[2], Charmie J Jain[3]

Assistant Professor, Information Science and Engineering, Sri Siddhartha Institute of technology, Tumakuru[1]

3rd Year, Information Science and Engineering, Sri Siddhartha Institute of technology, Tumakuru[2]

3rd Year, Information Science and Engineering, Sri Siddhartha Institute of technology, Tumakuru[3]

Abstract: Sorting algorithms are fundamental to computer science which is learnt in data structures of various programming language, facilitating the efficient organization of data. Understanding their intricacies and performance characteristics is crucial for students, educators, and developers. In this project, we present a Sorting Algorithm Visualizer, a web-based tool designed to provide an interactive platform for exploring various sorting algorithms.

The visualizer offers implementations of six popular sorting algorithms: Bubble Sort, Selection Sort, Insertion Sort, Merge Sort, Quick Sort, and Heap Sort. Through a user-friendly interface developed using HTML, CSS, and JavaScript, users can select algorithms, adjust parameters such as array size and sorting speed, and visualize the sorting process in real-time.

The visualizer dynamically updates the display as algorithms execute, providing step-by-step animations that helps in understanding the operations. In addition to this, the system calculates and displays the space and time complexities of each algorithm, empowering users to make informed comparisons.

With this architecture on clarity and interactivity, the Sorting Algorithm Visualizer serves as an educational resource for learners of all levels. By understanding the sorting algorithms, this project aims to inspire curiosity and facilitate learning in the field of computer science.

Keywords: Sorting algorithm, data structures, visualisation, sorting visualizer, processing.

I. INTRODUCTION

Sorting algorithms are fundamental to computer science. Efficiently sorting data is essential for a variety of applications, including database indexing, searching, and data analysis. Given the diversity and complexity of sorting algorithms, it can be challenging to grasp how they operate and to understand their respective efficiencies.

A sorting visualizer is a very useful educational tool that brings these algorithms to life, providing an interactive way to understand and compare different sorting techniques. By visualizing the sorting process, learners can see how elements are moved, compared, and swapped, making abstract concepts more understandable. This project leverages HTML, CSS, and JavaScript to create a dynamic visual representation of several popular sorting algorithms: Bubble Sort, Merge Sort, Quick Sort, Heap Sort, Insertion Sort, and Selection Sort. Additionally, it also displays the time and space complexities of each algorithm, along with a code snippet for better comprehension.

The primary objective of this sorting visualizer is to enhance the learning experience by allowing users to observe how different sorting algorithms operate on various datasets. By visually stepping through each algorithm, users can gain a deeper understanding of the mechanics behind each sorting method. The tool also serves as a valuable educational resource for students, educators, and anyone interested in algorithm design and analysis.

1. Interactive Visualization:
   - Real-time animation of sorting algorithms.
   - Step-by-step execution to show the sorting process.
2. Algorithm Selection:
   - Options to choose from Bubble Sort, Merge Sort, Quick Sort, Heap Sort, Insertion Sort, and Selection Sort.
3. Complexity Display:
   - Time complexity: Best, average, and worst-case scenarios.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 11, Issue 6, June 2024

DOI: 10.17148/IARJSET.2024.11634

- Space complexity: Memory usage for each algorithm.

4. Code Snippets:

- Clear and concise code snippets for each sorting algorithm to aid in understanding and implementation.

By incorporating these features, the sorting visualizer not only facilitates a better grasp of how each algorithm works but also provides a hands-on learning experience. Users can interact with the visualizations, experiment with different algorithms, and directly see the impact of changes in array size and sorting speed.

II. LITERATURE SURVEY

1. "Introduction to Algorithms" by Cormen et al.:

This seminal textbook is a cornerstone in algorithm education. Its comprehensive coverage of sorting algorithms, coupled with detailed explanations and time complexity analysis, provides a solid theoretical foundation. By studying this book, developers can gain insights into the inner workings of sorting algorithms, aiding in the creation of accurate visualizations and educational content.

2. "Algorithms" by Sedgewick and Wayne:

Known for its clarity and practicality, this book offers a wealth of sorting algorithms accompanied by Java code implementations. While primarily focused on Java, the clear explanations and visualizations can be adapted to JavaScript, enriching the sorting visualizer with well-documented algorithms and code examples.

3. "JavaScript Algorithms and Data Structures" by Bae:

Specializing in JavaScript implementations, this resource complements the previous two by providing optimized and best-practice JavaScript code for sorting algorithms. Its emphasis on implementation details ensures that the sorting visualizer's code is not only performant but also follows industry standards, enhancing its readability and maintainability.

4. "Sorting and Searching" by Knuth:

As part of Knuth's monumental "The Art of Computer Programming" series, this book offers unparalleled depth in sorting algorithm theory and design principles. By understanding the theoretical underpinnings elucidated in this work, developers can create a visualizer that not only demonstrates algorithmic mechanics but also provides insights into algorithm design rationale and efficiency analysis.

5. Existing Sorting Visualizers:

Analysing existing projects on platforms like GitHub and CodePen provides practical insights into user interface design, animation techniques, and additional features. By studying successful implementations, developers can incorporate best practices and innovative ideas, ensuring that the sorting visualizer is intuitive, engaging, and informative for users.

III.PROPOSED SYSTEM

The proposed system, SORTIFY, is designed to create an interactive and educational tool for visualizing sorting algorithms. Using HTML, CSS, and JavaScript, SORTIFY will provide real-time animations of popular sorting algorithms like Bubble Sort, Merge Sort, Quick Sort, Heap Sort, Insertion Sort, and Selection Sort. Users can select an algorithm, adjust array size and sorting speed, and observe the sorting process dynamically. The system will also display the time and space complexities of each algorithm and include code snippets for better understanding. SORTIFY aims to enhance learning by making algorithmic concepts accessible and engaging through interactive visualization. This will help all the user with no age limits and without actually knowing the algorithm in deep also.

IV.SYSTEM ARCHITECTURE

The system architecture for the sorting visualizer project is illustrated in the provided diagram. It is structured to facilitate a clear, interactive, and educational experience for users learning about sorting algorithms. Here's a breakdown of each component and its functionality:

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 11, Issue 6, June 2024

DOI: 10.17148/IARJSET.2024.11634

Control Panel

- Algorithm Selection: Users can choose from a variety of sorting algorithms, including Bubble Sort, Selection Sort, Insertion Sort, Merge Sort, Quick Sort, and Heap Sort.
- Adjust Size of Array: Users can adjust the size of the array to be sorted, allowing them to see how different algorithms handle varying amounts of data.
- Speed of Sorting: Users can control the speed of the sorting visualization, enabling them to slow down or speed up the process for better understanding.

Visualization Area

- Visual Sorting Process: This area displays the real-time animation of the chosen sorting algorithm. Users can visually follow the steps of the algorithm as it sorts the array.
- Time and Space Complexity: Detailed information about the time complexity (best, average, and worst-case scenarios) and space complexity of the selected algorithm is displayed, helping users understand the efficiency and performance implications.

Code Snippet Display

- Code Snippet of Particular Sorting Algorithm: This section provides the code snippet for the currently selected sorting algorithm. The code is presented in a clear and concise format, aiding in the understanding and implementation of the algorithm.

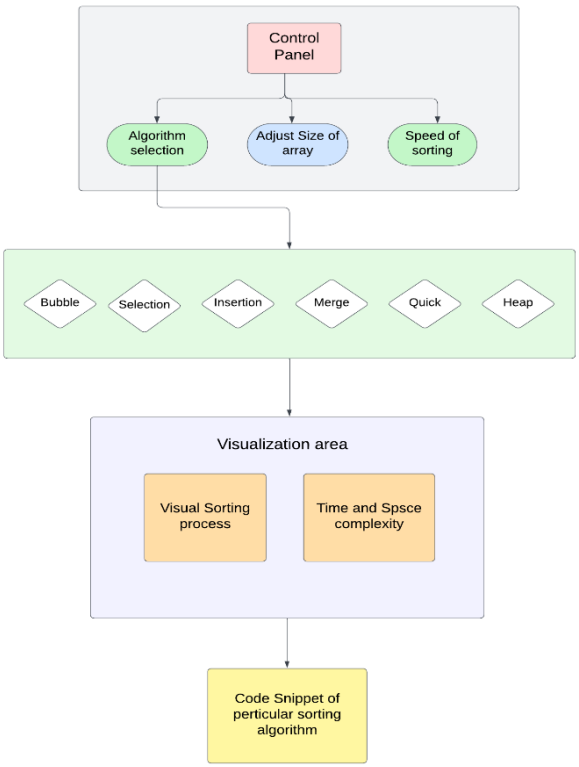

Fig. 1 Sortify – System Architecture

This flowchart outlines the user interface for a sorting algorithm visualization tool, including a control panel for selecting the algorithm, adjusting array size, and sorting speed.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 11, Issue 6, June 2024

DOI: 10.17148/IARJSET.2024.11634

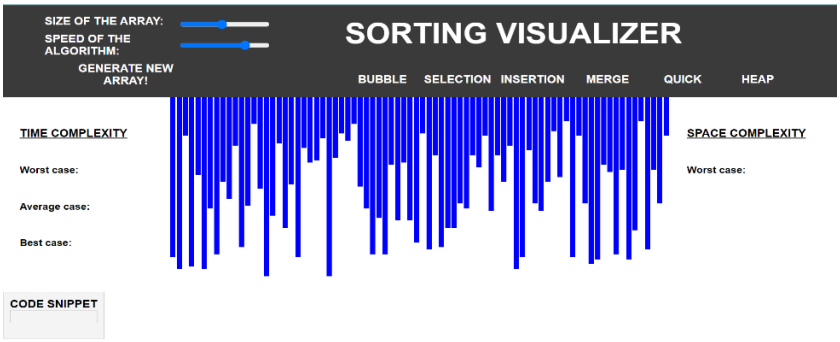

Fig. 2 Unsorted Visualizer

A visualizer picture description of "unsorted" might depict a chaotic jumble of elements, such as colourful blocks or shapes scattered randomly. The lack of order conveys a sense of disarray and unpredictability, emphasizing the absence of structure or organization.

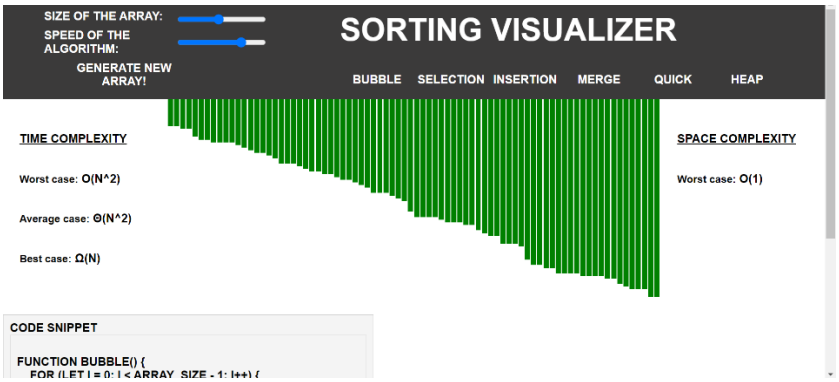

Fig. 3 Sorted Visualizer

A sorted visualizer picture displays a sequence of bars or lines arranged in ascending or descending order, visually representing the sorted elements. Each element's height or position correlates with its value, illustrating the sorting process.

CODE SNIPPET

```
FUNCTION BUBBLE() {
    FOR (LET I = 0; I < ARRAY_SIZE - 1; I++) {
        FOR (LET J = 0; J < ARRAY_SIZE - I - 1; J++) {
            IF (DIV_SIZES[J] > DIV_SIZES[J + 1]) {
                // SWAP DIV_SIZES[J] AND DIV_SIZES[J + 1]
                LET TEMP = DIV_SIZES[J];
                DIV_SIZES[J] = DIV_SIZES[J + 1];
                DIV_SIZES[J + 1] = TEMP;

                // UPDATE THE VISUAL ELEMENTS
                DIVS[J].STYLE.HEIGHT = DIV_SIZES[J] + '%';
                DIVS[J + 1].STYLE.HEIGHT = DIV_SIZES[J + 1] + '%';
            }
        }
    }
}
```

Fig. 4 Code Snippet of Bubble Sort

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 11, Issue 6, June 2024

DOI: 10.17148/IARJSET.2024.11634

The provided code snippet is an implementation of the Bubble Sort algorithm that also updates the height of visual elements to reflect the changes in the array.

V. CONCLUSION

SORTIFY represents a milestone in educational technology, offering a straightforward yet powerful platform for understanding sorting algorithms. Built with HTML, CSS, and JavaScript, the visualizer provides an intuitive interface where users can interactively explore sorting techniques. By adjusting array sizes and algorithm speeds, users gain practical insights into algorithmic behaviour. The inclusion of code snippets further enhances comprehension, allowing users to grasp the underlying logic of each algorithm in a clear and accessible manner.

Looking forward, SORTIFY remains committed to simplicity and effectiveness. Plans for future development include expanding the range of sorting algorithms available and refining the user experience to ensure accessibility for learners of all levels. By remaining responsive to user feedback and continuously iterating on the platform, SORTIFY aims to continue serving as a valuable resource for those seeking to understand and master sorting algorithms in a straightforward and engaging manner.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the Department of Information Science and Engineering at the Sri Siddhartha Institute of Technology (SSIT), Tumakuru, Karnataka, for providing the necessary resources and support for this research. We extend our appreciation to our colleagues and faculty members for their insightful feedback and encouragement throughout the project. The proposed system targets in building an interactive and educational tool for visualizing sorting algorithms such as Bubble sort, Insertion sort, Selection sort, Merge sort, Quick sort, Heap sort as well as displaying time and space complexities and code snippet of each algorithm.

REFERENCES

[1]. “Introduction to Algorithms” Authors: Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford SteinY.
[2]. “Data Structures and Algorithms Made Easy” Author: Narsimha Karumanchi
[3]. “Learning Web Design: A beginner’s guide to HTML, CSS, JavaScript, and Web Graphics” Author: Jennifer Niederst Robbins
[4]. “JavaScript and JQuery: Interactive Front-End Web Development” Author: Jon Duckett.
[5]. “Software Visualization” Editor: Stephan Diehl