International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 9, pp 45-49, September-2024

https://doi.org/10.47001/IRJIET/2024.809004

Data Structure Visualizer

[1]Siddhi Kengar, [2]Yuganti Kumkar, [3]Megha Shinde, [4]Swathi Pepeti, [5]Snehal Bawkar

[1,2,3,4,5]Student, Genba Sopanrao Moze College of Engineering, Balewadi, Pune, Maharashtra, India

***Abstract* - Effective algorithm visualization in computer science education greatly enhances the level of understanding and interest in the subject. This paper presents a development and evaluation of a web-based visualizer of data structures intended for learning sorting algorithms. Our system allows interactive and dynamic visualization of various sorting techniques, including Bubble Sort, Merge Sort, Quick Sort, and Heap Sort. We provide a real-time graphical display for the user to view step-by-step executions of such algorithms. This provides insight into their operational mechanics and performance characteristics. We present the architecture of the visualizer, methodologies for algorithm visualization, and some feedback from users during testing phases. Both qualitative and quantitative metrics are used for assessing the improvement in algorithmic understanding. Our findings indicate that the visualizer significantly enhances students' understanding of the concepts in sorting algorithms and that it is indeed a very important tool for computer science educators and learners alike.**

Keywords: Data Structure, Visualizer, Visualization in computer science, Dynamic visualization.

I. INTRODUCTION

Sorting algorithms form a vital part of computer science; they find applications right from data processing to real-time systems. Traditionally, most methods lack the potentials to clearly explain their dynamic nature and operational details. In view of this gap, we are developing a web-based visualizer for data structures that targets sorting algorithms like Bubble Sort, Merge Sort, Quick Sort, and Heap Sort. It provides an interactive, real-time graphical representation that is useful in understanding step-by-step processes of algorithms. By making this visualizer more intuitive and engaging, it enhances the learning of concepts regarding sorting algorithms and will definitely be one of the helpful tools for instructors and learners alike in computer science education.

1.1 Project Aim and Objective:

This project will implement a web-based visual data structure visualizer with the purpose of improving understanding and the educational process concerning sorting algorithms through interactive and dynamic visualization. Objectives:

1. Design and Development: Develop an intuitive web-based platform that will facilitate the illustration of various sorting algorithms, namely Bubble Sort, Merge Sort, Quick Sort, and Heap Sort. The user must be allowed to interactively engage in the sorting processes in real time using the platform.
2. Interactive Visualization: Embed interactive graphical elements to enable users to go step-by-step through every stage of the sorting algorithms by underlining the most important operations: comparisons, swaps, and movements of data.
3. Learning Efficiency: Establish the efficiency of the visualizer in comprehensively learning different sorting algorithms through tests and solicitation of user feedback. Perform qualitative and quantitative analysis of several metrics to establish the impact on learning outcomes.
4. Usability and Accessibility: The visualizer needs to be user-friendly and accessible to meet a wide range of users' needs, from novice to advanced. The interface therefore needs to be structured in a way that its use and interaction are relatively easy.

1.2 System Objectives:

The system shall be designed to provide the primary function-a web-based, interactive visualizer capable of dynamically displaying real-time graphs of the following sorting algorithms: Bubble Sort, Merge Sort, Quick Sort, and Heap Sort. This system intends to fortify the user's understanding by allowing them to visually trace along and interact with these sorting processes. The key features are step-by-step execution, immediate visual feedback on operations, and intuitive controls that make the activities highly engaging yet easy to perceive by the user.

Meanwhile, the system is designed to be user-friendly and accessible across devices and browsers, hence a DDA-compliant system that should find application in serving a wide range of learning needs.

Accompanied by detailed documentation and support, the visualizer tries to support both learners and educators in teaching and understanding sorting algorithms more effectively in order to improve educational outcomes and foster a more interactive learning environment.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 9, pp 45-49, September-2024

https://doi.org/10.47001/IRJIET/2024.809004

1.3 Background of Project:

Sorting algorithms are the backbone of computer science, very important in efficiently processing and organizing data. It plays an integral part in most computing tasks, from database management systems to real-time systems. However, the abstract nature of sorting algorithms most often makes the intricacies of various sorting algorithms hard to grasp with theoretical explanations and static diagrams.

Recent developments in educational technologies have created new opportunities to address these challenges. These come in the form of interactive visualizations that allow a viewer to experience algorithmic processes in real time through graphical means. Thus, observing sorting algorithms in interaction will convey an intuitive sense of their operating principles and relative efficiencies more clearly to the learner. This project continues this educational trend by developing a web-based data structure visualizer destined for sorting algorithms. In such a way, the tool will narrow the gap between theory and practice, make the learning process more interactive, and easy for both students and educators alike.

II. COMPONENTS

2.1 Software Components for Processing the System:

1. Frontend Components:

Mainly, HTML, CSS, and JavaScript are the foundational technologies availed. Your system design focuses on browser-based processing without an actual need for a server-side backend.

2. HTML (User Interface Structure):

Defines the structure of a web page, which includes the structure of the buttons and input fields for algorithm selection and the graphical canvas to render the data being sorted.

Makes a simple interface to choose any of the sorting algorithms and array size with a button to begin the operation of sorting.

3. CSS (Styling and Basic Animation):

Styles the interface and visualization elements, such as a bar and block, to make it look good.

Basic animations-color change, hover, layout adjustment, etc.-should be handled inside the visualizer.

4. Javascript [Core Logic and Interaction]:

Implementation of Sorting Algorithms: Java functions implementing each sorting algorithm, for example, Bubble Sort, Quick Sort, etc.

Data Generation: Functions that generate random arrays and/or take input from the user for an array to sort.

5. Version Control & Collaboration:

GitHub Repository: GitHub handles the version control to make it easy for collaborators and to deploy easily. In case it is required, a website can also be hosted on GitHub pages.

III. METHODOLOGY

The following will be the steps for the Data Structure Visualizer project:

1. Planning and Requirement Analysis:

Define the objective: To design a web-based sorting algorithm visualizer using HTML, CSS, and JavaScript.

2. Design Phase:

a. Design a simple user interface with controls for users to select any algorithm and visualization elements-data representation with bars or blocks.

3. Development Phase:

HTML: Give the structure to the webpage.

CSS: Style the interface by adding basic animations.

JavaScript: Sorting algorithms will be implemented along with real-time data visualization and user interaction management..

4. Deployment Phase: Version control is done using GitHub. For hosting to go online, use GitHub Pages. 6. Documentation Phase: This will involve thorough documentation on the code and features of the projects. This will get the project ready for future reference or presentation.

IV. RESULT

The Data Structure Visualizer project is an exhaustive, interactive, web-based system that provides help to understand various data structures and algorithms. Here are the contents of the homepage of this website:

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 9, pp 45-49, September-2024

https://doi.org/10.47001/IRJIET/2024.809004

4.1 Data Structures Visualizations:

4.1.1 Arrays: Array visualizations have been done for linear search, binary search, and sorting algorithms. A user can get a feel as to how elements are searched for, using different search algorithms and how elements are sorted using different sorting algorithms.

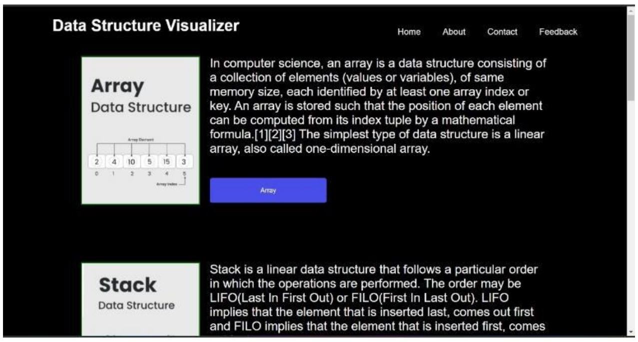

Figure 1

4.1.2 Stacks: Some basic visualizations present stack operations, push and pop.

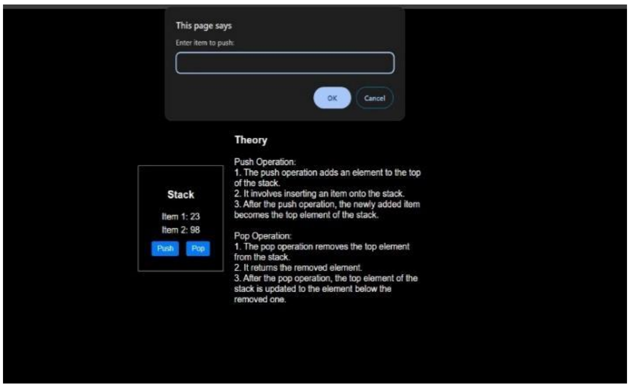

Figure 2

4.1.3 Queue: Offers visualizations of the operations in a queue, enqueue and dequeue.

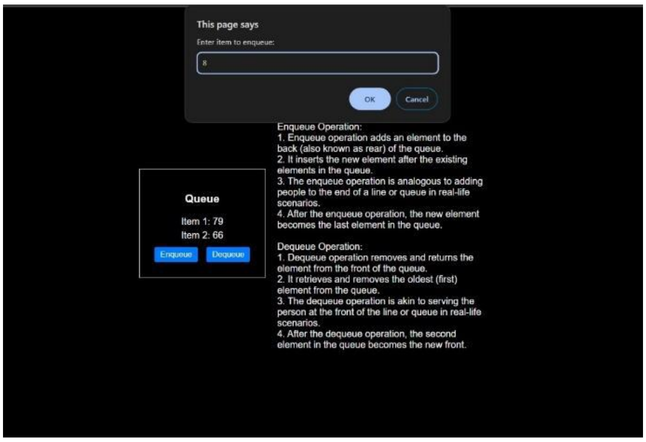

Figure 3

4.1.4 Linked Lists: Offers a view to the structure and manipulation of the linked list, including insertions and deletions.

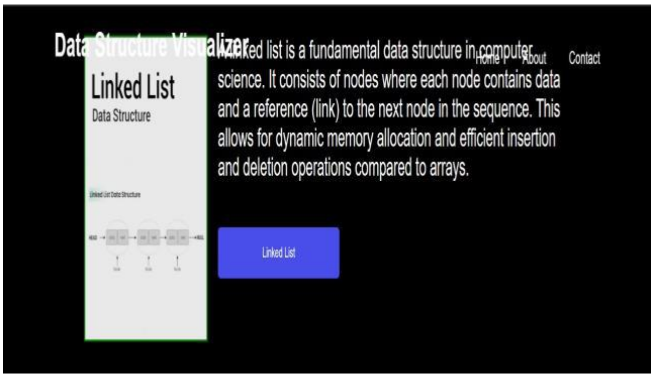

Figure 4

4.1.5 Trees: Offers visualization of tree structures with common operations such as insertions and deletions. Common traversal methods in a tree.

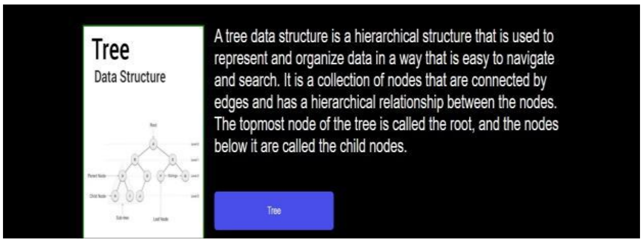

Figure 5

4.1.6 Graphs: It includes basic visualizations of graph data structure and standard algorithms such as DFS and BFS.

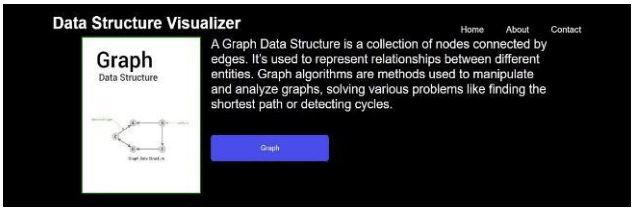

Figure 6

4.2 Algorithm

4.2.1 Array: Offers linear search and binary search in the array.

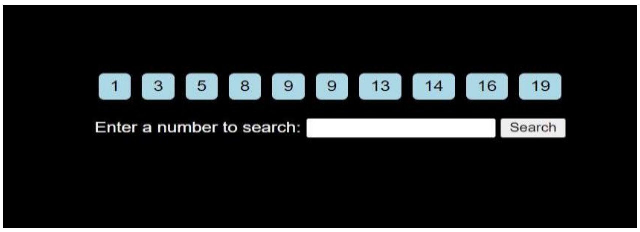

Figure 7

ISSN (online): 2581-3048
Volume 8, Issue 9, pp 45-49, September-2024
https://doi.org/10.47001/IRJIET/2024.809004

4.2.2 Video Lectures on Sorting Algorithms: It shows a variety of sorting techniques, including Bubble Sort, Merge Sort, Quick Sort, and many others. In each one of them, it visually demonstrates step by step how a certain algorithm will sort an array.

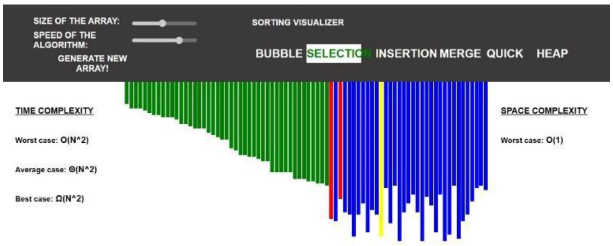

Figure 8

V. CONCLUSION

The Data Structure Visualizer project effectively creates an engaging web-based tool for better comprehension of the most important data structures and algorithms. This tool will let users visualize the workings of these structures with interactive visualizations for arrays, stacks, queues, linked lists, trees, and graphs. Furthermore, it provides real-time animation of sorting algorithms and basic operations, enhancing the educational experience for users by showing, in action, this generally complex process.

The project contributes to the learning and teaching of basic computer science interactively. It makes the study of data structures and algorithms intuitive by putting together detailed visualization with interaction.

VI. FUTURE SCOPE

The Data Structure Visualizer project has lots of possible extensions and improvements. Among the foreseen enhancements are the following:

1. Extended Data Structures and Algorithms: Additional data structures will be visualized, including hash tables, AVL trees, or B-trees, together with more sophisticated algorithms, for example, Dijkstra's or A* for graph processing.
2. Interactive Capabilities: Allowing the user to interact directly with the visualizations themselves-be it by inputting their own data for custom arrays or graphs, changing visualization options (e.g., animation speed), or manually being able to step through algorithms.
3. Back-End Development: Incorporate a back-end structure where user preferences or progress could be stored so that users can create personalized learning spaces and return to any previous visualizations.
4. Visualization: Enhance the visualizations using more detailed graphics and animations. You could use WebGL to show 3D visualizations or add more learning content, like the complexity analysis of algorithms.
5. Cross-browser compatibility: Compatibility should ensure the best experience from any kind of device/browser, adding mobile devices support.

REFERENCES

[1] Sedgewick, R. (1998). Algorithms in Algorithms C++, parts 1-4: fundamentals, data structure, sorting, searching. Pearson Education. Kerren et, "Algorithm Animation".

[2] lladi Sutopo, "SELECTION SORTING ALGORITHM VISUALIZATION USING FLASH", The International Journal of Multimedia & Its Applications (UMA) Vol.3, No.1, February 2011[2] https//aircconline.com/ijma/V3N1/3111ijma03pdf.

[3] Reif and T. Orzhovacki, "VISA: Visualization of Sorting Algorithms", 2012.

[4] Slavomir Simonak, "Increasing the Engagement Level in Algorithms and Data Structures.

[5] Course by Driving Algorithm Visualizations" September 2020, Informatica 44(3).

[6] Hungarian (Kükülömenti legényes) folk dance showing sorting by dance.

AUTHORS BIOGRAPHY

Siddhi Kengar,
Pursuing B.E. AI&DS at Genba Sopanrao College of Engineering, Balewadi, Pune, Maharashtra, India.

Yuganti Kumkar,
Pursuing B.E. AI&DS at Genba Sopanrao College of Engineering, Balewadi, Pune, Maharashtra, India.

Megha Shinde,
Pursuing B.E. AI&DS at Genba Sopanrao College of Engineering, Balewadi, Pune, Maharashtra, India.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 9, pp 45-49, September-2024

https://doi.org/10.47001/IRJIET/2024.809004

Swathi Pepeti,
Pursuing B.E. AI&DS at Genba Sopanrao College of Engineering, Balewadi, Pune, Maharashtra, India.

Snehal Bawkar,
Pursuing B.E. AI&DS at Genba Sopanrao College of Engineering, Balewadi, Pune, Maharashtra, India.

Citation of this Article:

Siddhi Kengar, Yuganti Kumkar, Megha Shinde, Swathi Pepeti, & Snehal Bawkar. (2024). Data Structure Visualizer. *International Research Journal of Innovations in Engineering and Technology - IRJIET*, 8(9), 45-49. Article DOI https://doi.org/10.47001/IRJIET/2024.809004

****

