STM JOURNALS

International Journal of
VLSI Circuit Design & Technology

Volume 2, Issue 1, 2024
DOI (Journal): 10.37591/IJVCDT
(Jan-June)

https://journals.stmjournals.com/ijvcdt

Review IJVCDT

Visualizing Complexity: Navigating Algorithms with Algorithm Visualizer

Akshada Gaikwad[1], Sayali Jagtap[2], Priyanka Kamble[3], Rashmi Shinde[4], N.M. Dimble[5]

Abstract

We think that studying algorithms may be amusing and exciting. Many students are currently struggling, but we hope to change that. Our strategy is to emulate gaming while learning. We're building hands-on learning exercises, such as mazes and patterns, to convey key concept. To help students understand how algorithms function in practice, we also use some stunning visualizations. Three major categories of algorithms are under our purview: sorting, pathfinding, CPU scheduling, machine learning, and encryption techniques. They are crucial and frequently applied in computer science. Not only is learning algorithms to be easier, but we also want to make it fun. Furthermore, this will not only help students but also provide educators with fresh approaches to imparting these ideas in a way that is more engaging and intelligible. These make it easier for the eyes to understand how algorithms operate. Observing something in action visually retains information better in our thoughts. This facilitates learning and makes it more pleasurable as well. It lays a solid foundation for further discussions of more complicated concepts by providing us with an understandable image of how things like data structures are put together. We're employing animated images to add even more spice to the learning experience and make it much more interactive and engaging.

Keywords: Algorithms, ML, encryption algorithms, pathfinding, sorting, CPU scheduling, and visualization

INTRODUCTION

Certain topics in computer science are like the foundation for everything else you study. “Design and Analysis of Algorithms” is one of those essential topics that deals with algorithms. The truth is, though, that many individuals have a really difficult time understanding algorithms. Motivated by the notion that observing something in action facilitates a deeper understanding, we made the decision to develop a platform that brings together all the key algorithms and illustrates them with engaging interactive elements and graphics.

***Author for Correspondence**
Rashmi Shinde
E-mail: shinderashmi2002@gmail.com

[1-4]Student, Department of Computer Engineering, Rajgad Dnyanpeeth’s, Shri Chhatrapati Shivajiraje College of Engineering, Dhangawadi, Pune, Maharashtra, India
[5]Professor, Department of Computer Engineering, Rajgad Dnyanpeeth’s, Shri Chhatrapati Shivajiraje College of Engineering, Dhangawadi, Pune, Maharashtra, India

Received Date: May 15, 2024
Accepted Date: May 23, 2024
Published Date: July 10, 2024

Citation: Akshada Gaikwad, Sayali Jagtap, Priyanka Kamble, Rashmi Shinde, N.M. Dimble. Visualizing Complexity: Navigating Algorithms with Algorithm Visualizer. International Journal of VLSI Circuit Design & Technology. 2024; 2(1): 21–26p.

Visualizations are narratives with graphics that illustrate an algorithm's steps of operation. People will find it simpler to understand how the algorithm is operating as a result. What's the best thing, then? Proceed at your own speed. As a novice, you can go cautiously. Things can move more quickly if you're more experienced. But there's still more! To make learning feel like play, we also included entertaining activities, such as games with mazes and patterns. We include content for novices and children as well, so it's not just for specialists. We also provide you with estimates of the time and space required if you're serious about algorithms. We set out on an

engrossing journey into the realm of Algorithm Visualizer, investigating its functionalities, attributes, and the revolutionary influence it has on comprehending and becoming an expert in algorithms. Algorithm Visualizer provides a distinctive and priceless learning experience for hobbyists, students, and professionals alike, from its user-friendly interface to its dynamic visualization approaches.

As of right now, we have visualizations for common activities such as pathfinding, sorting, scheduling, prediction, and computer encryption. However, we won't stop there. Our objective is to cover every possible combination of algorithms that exists. Thus, our platform has something to offer everyone, regardless of experience level [9–10].

LITERATURE SURVEY

1. Daniela Borissova and Ivan Mustakerov [1] have developed an e-learning tool focused on visualizing algorithms for finding shortest paths. This tool facilitates the creation, editing, and saving of graph structures while visualizing the execution steps of these algorithms. It serves as a versatile resource intended to complement traditional classroom instruction or function independently as a standalone application. The tool is designed to accommodate various algorithms for determining shortest paths, enhancing its utility and comprehensiveness in educational settings.
2. Neetu Goel et al. [2] present a paper titled “A Comparative Study of CPU Scheduling Algorithms,” which introduces a state diagram comparing various CPU scheduling strategies. This diagram facilitates understanding of system operations and decisions regarding program execution on the CPU. The study aims to identify the most effective CPU scheduler by evaluating and comparing different algorithms for optimal process scheduling efficiency.
3. R. Mavrevski et al [3]. exemplify the application of Dijkstra's algorithm to solve the classic problem of finding the shortest path between two vertices in a connected, undirected graph. This problem is frequently encountered and addressed in competitions like the annual International Olympiad in Informatics, showcasing practical implementations and insights derived from algorithmic solutions.
4. Brian Faria et al. [4] explore the educational benefits of animated sorting algorithms in computer science courses. Their study visualizes Selection Sort, Bubble Sort, Insertion Sort, and Merge Sort using a custom web-based animation tool. This tool allows users to select an algorithm and data ordering technique, presenting data as interactive bar graphs. Users can then navigate through the animations automatically or manually, enhancing their understanding of sorting processes
5. Mykhailo Klunko et al. [5] aimed to develop user-friendly software for enhancing understanding of typical sorting algorithms. The primary goal was to create a tool that prioritizes optimal user experience. This demonstration program ensures ease of use and accessibility, allowing users to apply each sorting method to their own data, thereby maximizing educational engagement and learning outcomes.
6. Pedro Moraes et al. [6] underscore the foundational role of Data Structures and Algorithms (DSA) in software development. They highlight the challenge in teaching and comprehending these concepts due to their abstract nature. Commonly, instructors employ methods such as slides and whiteboard sketches to illustrate the behaviors and operations of DSAs, aiding students in visualizing and understanding these fundamental principles.
7. Qiang Gao et al. [7] introduce and analyze foundational concepts in computational complexity. They explore essential aspects such as complete problems in time and space complexity, supported by illustrative examples. Additionally, the paper provides a detailed analysis of the relationships among various complexity classes.
8. Anushka Dhar, Barnini Goswami, et al [8]., employed a visualization tool to explore Sorting and Path Finding Algorithms. This application enables users to create mazes and patterns to facilitate comprehension. Users can select from a range of sorting and path-finding algorithms and visualize them at adjustable speeds. Leveraging React.js and JavaScript, known for their

versatility in creating intuitive user interfaces, the program was meticulously developed to enhance user interaction and learning experience.

PROPOSED METHODOLOGY

The following constitute the main elements of our work:

- Algorithms for sorting.
- Welcome page

These algorithms include hashing, scheduling, machine learning, pathfinding, and scheduling see in Figure 1.

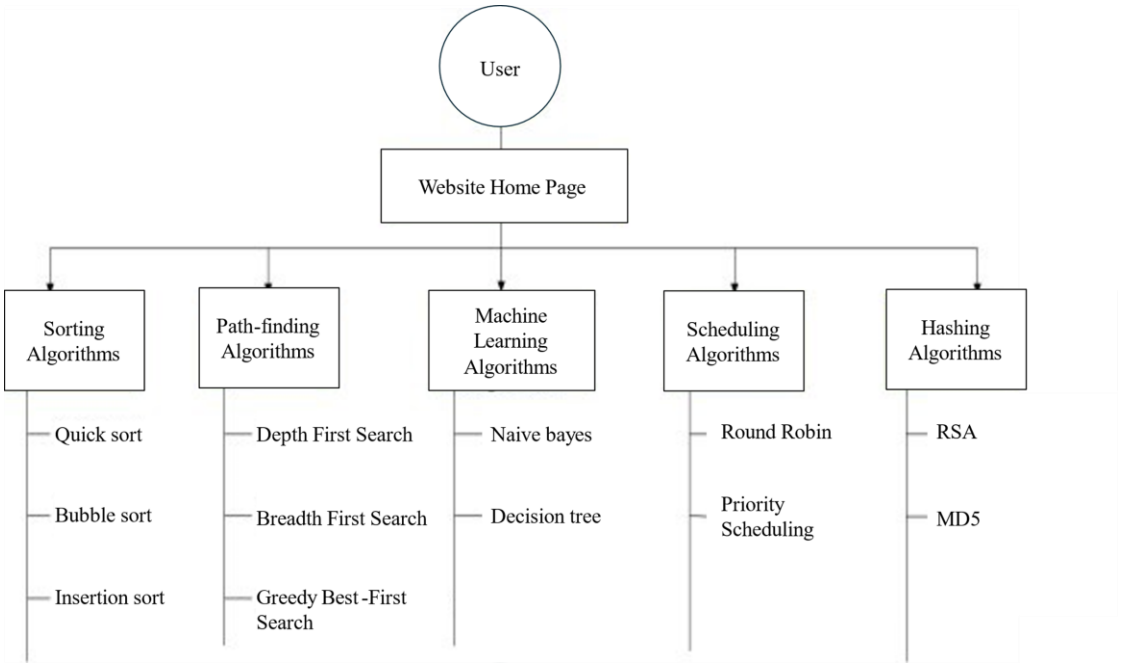

Figure 1. Proposed System.

System Architecture

We have emphasized a graphical interface that uses many colors to distinguish between distinct aspects, such as walls, nodes, algorithms, and unvisited nodes, with unvisited nodes displayed in white, to guarantee that our algorithm visualization application offers a seamless user experience. Recognizing that the user is the center of our application, we have concentrated on creating a user-friendly and captivating experience.

Our application's system architecture is shown in Figure 2. centered on giving consumers a smooth, visually intuitive experience. Using a graphical user interface, we have carefully applied a color-coding scheme to differentiate objects, including walls, nodes, algorithms, and unvisited nodes (shown as white). This deliberate use of color improves user understanding and makes complex systems easier to visualize. Our design philosophy places a strong emphasis on making the user experience (UX) the center of attention for our product. Understanding the importance of a positive user journey, our goal was to increase the application's overall impact. We chose lightweight frameworks and scripting languages because our relationships and interactions are simple. The selection of the architecture described in our methodology section was the result of a thorough examination of the designs that were already available online. Making this strategic choice guarantees that application effectiveness is maximized while industry standard practices are followed. We are committed to providing an optimized and significant solution, which is demonstrated by our use of lightweight technology, a well-defined architecture, and a user-centric approach.

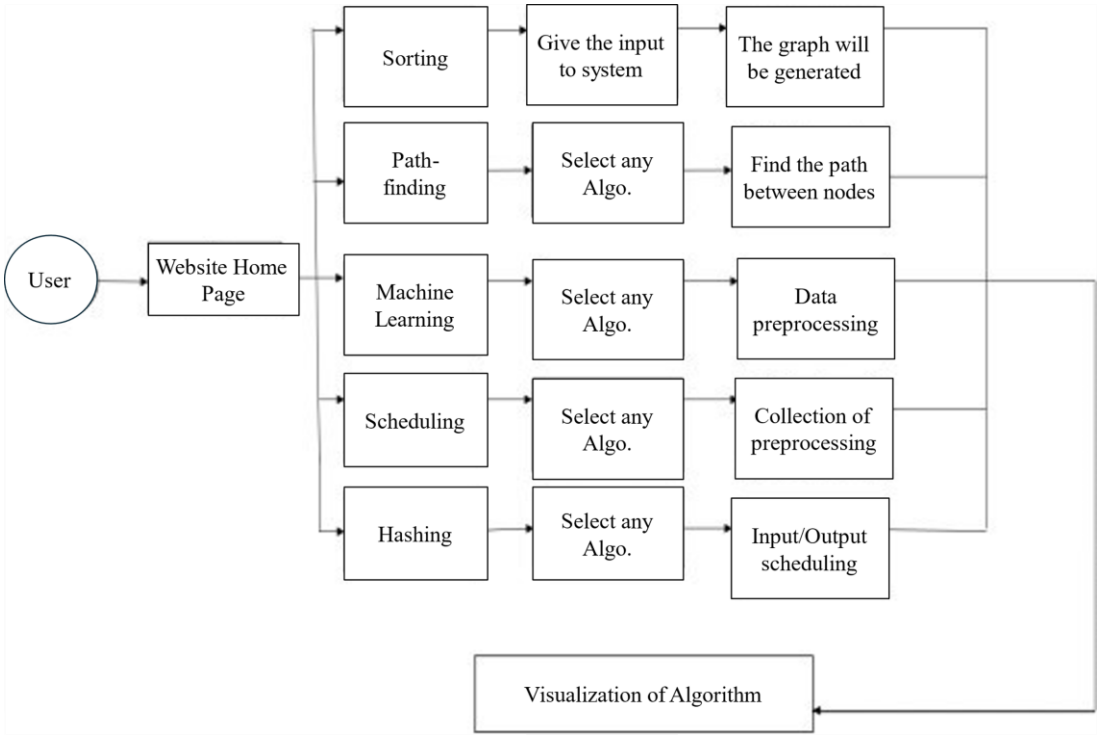

Figure 2. System Architecture.

Workflow

Users can explore numerous domains from the welcome page with the goal of comprehending and navigating through various algorithms:

Examine techniques and algorithms created to arrange data alphabetically. This will provide you insights into the most effective ways to arrange data.

Explore Path-Finding methods: Explore methods designed to determine the best paths in a variety of circumstances, including environments with grids or graphs. Users can better understand the workings of pathfinding algorithms such as Scrappy Best-First Look, Depth First Search (DFS), and Breadth First Search (BFS) with the use of visualizations.

Discover an overview of the algorithms used in machine learning, such as Naive Bayes classifiers and Decision Trees. It is easier to comprehend how these algorithms categorize and forecast results when visual representations are used.

Program Algorithms: Examine the basic algorithms used in programming assignments to get knowledge of the key algorithms in software development.

Learn about the cryptographic hashing methods, such as RSA and MD5, that are used to ensure the security and integrity of data. Hashing and its uses for safe data storage and transfer are demonstrated through visualizations.

Users can effectively understand the intricacies and applicability of these algorithms by utilizing the interactive visuals and explanatory tools included in each domain.

CONCLUSION

In conclusion, we personally recognized widespread problems with conventional teaching approaches and worked to offer an additional remedy. Our approach tackles the difficulty that arises

from inadequate teacher-student communication while attempting to understand principles of Algorithm Visualizers. In what way might our solution be helpful?

A. *Visual Learning:* We make use of the visual aids—images, films, and animations—that are proven to improve learning.
B. *User-Friendly:* Our program is made to be simple to use and can be utilized by people of all ages.
C. *User Engagement:* We guarantee user engagement and satisfaction by integrating real-world algorithm implementations.
D. *Time Complexity Comparison:* To compare algorithms in the same domain, our program has a time complexity feature.
E. *Extensive Coverage:* For algorithm lovers, we provide a one-stop shop of frequently used algorithms, such as sorting, path-finding, and CPU scheduling.

Future Scope

Our current application effectively illustrates the time complexity of fundamental algorithms like sorting, pathfinding, and CPU scheduling, as well as patterns and mazes.

To further improve the visualizations and make them even more captivating and intelligible, we intend to include several new features in the future:

A. *Tree-based Visualization of Algorithms:* In textbooks, tree data structures are used to illustrate a lot of algorithm topics. In our application, we will introduce tree-based visualization as a means of bridging the gap between traditional learning methods and e-learning. Users will gradually switch from traditional learning methods to dynamic algorithm visuals as they go through the tiers.
B. *Line-by-Line Analysis Aids in Visualization Code Execution:* To help with understanding, we'll provide a feature that lets users see algorithms side by side with the appropriate code, shown line by line. This two-view configuration will Assist students in comprehending the relationship between code execution and visual representations to improve their comprehension and exam readiness.
C. *Pause and Play Functionality:* Users will be able to pause algorithm visualizations and pick up where they left off, much like they may pause and resume videos. This feature guarantees continuous learning, letting users go back over material as needed without having to start over.
D. *Modeling Real-world events:* By modeling real-world events, we hope to highlight the practical applicability of algorithms. As an illustration: Delivery and Logistics Management: Users can investigate delivery optimization strategies, benefiting both students and
E Helping businesses in comprehending the use of resources and prompt customer service.

Geolocation and navigation maps in real-time: This feature will pique users' curiosity about the uses of algorithms in daily life by showcasing how shortest route-finding algorithms enable navigation apps.

F. *Smooth transfer to college study:* Our features improve students' productivity and knowledge by facilitating a seamless transfer from school to college-level study.

To put it briefly, our solution seeks to enhance students of all levels' comprehension of concepts, develop a deeper grasp of algorithms, and facilitate smooth educational advancement.

REFERENCES:

1. Borissova D, Mustakerov I. E-learning tool for visualization of shortest paths algorithms. Trends Journal of Sciences Research. 2015 Sep 30;2(3):84-9.
2. Neetu Goel and Dr. R.B. Garg (2012) 'A Comparative Study of CPU Scheduling Algorithms.', International Journal of Graphics and Image Processing,

3. Mavrevski R, Traykov M, Trenchev I. Finding the shortest path in a graph and its visualization using C# and WPF. Int. J. Electr. Comput. Eng. 2020 Apr 1;10(2):2054-9.
4. Klunko M, Klunko M. Visualization of Sorting Algorithms. Brian Faria (2017) 'Visualizing sorting algorithms', Rhode Island College
5. Moraes P, Teixeira L. Willow: A tool for interactive programming visualization to help in the data structures and algorithms teaching-learning process. In Proceedings of the XXXIII Brazilian Symposium on Software Engineering 2019 Sep 23 (pp. 553-558).
6. Gao Q, Xu X. The analysis and research on computational complexity. InThe 26th Chinese Control and Decision Conference (2014 CCDC) 2014 May 31 (pp. 3467-3472). IEEE.
7. Goswami B, Dhar A, Gupta A, Gupta A. Algorithm Visualizer: Its features and working. In2021 IEEE 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON) 2021 Nov 11 (pp. 1-5). IEEE.
8. Shaffer CA, Cooper ML, Alon AJ, Akbar M, Stewart M, Ponce S, Edwards SH. Algorithm visualization: The state of the field. ACM Transactions on Computing Education (TOCE). 2010 Aug 1;10(3):1-22.,
9. Kucera L. Visualization of Abstract Algorithmic Ideas. InCSEDU (1) 2018 (pp. 497-504)

