Interactive Online Platform for Visualizing an Algorithm

Dipak Ghodekar[1], Prof. Meghraj Patil[2], Pratik Das[3], Avishkar Shetkar[4], Saurabh Potdar[5]

[1,2,3,4,5]Information Technology/Genba Sopanrao Moze College of Engineering/India

ABSTRACT

As the distance and online learning is going on integrated platforms are required for students learning. our website is a platform that aims on "Algorithm Visualization", which will ease the learning process and help to understand algorithms in better way. This website includes various other functionality that is Assessment, algorithm explanation and visualization. It also checks the knowledge of the learner, on the basis of the result of assessment tests. This website is very useful for computer science student and teachers for learning data structures and algorithms. The main of this project is to make better understanding of data structures and algorithm. This website also provides the knowledge about data structure array, stack, queue, and linked list as well. Thus, our project provides better understanding of data structures and algorithms.

Keywords: Algorithm, Algorithm Visualizer, data structures, array, stack, queue, linked list

INTRODUCTION

Data Structures and Algorithms (DSA) being the necessity of Computer Engineering, their conceptual understanding is of most important. By various surveys, it was observed that if topics are explained and demonstrated graphically it becomes easy to that user to grasp that topic. Although many Algorithms Visualization tools are available online, most of them are not effective in helping students to learn algorithm and data structures. The main reasons for this may be the poor design of the tools to fulfill the learners' needs in terms of usability and accessibility. The original motive of Algorithm Visualizer was an effort to aid novices attempting to understand standard computing algorithms. The instructional process in traditional classes was often without the opportunity to experiment and explore a concept in other dimensions. Its main aim is to visualize the algorithms of data structure. visualization of algorithms will be helpful for student to understand the algorithms in a precise and accurate manner. With his application, students can learn and code various algorithms visually. User can track his/her progress through assessment.

The main objective of this project is to help beginners to be able to visualize the basic algorithms and get a better understanding of the underlying operations. Implementing algorithms is a key step for the transfer of algorithmic technology, which often requires a high- level of expertise, to different and broader communities. Algorithm implementations also used for its effective deployment in industry and real applications.

LITERATURE REVIEW

In 2005 JHAVE supporting algorithm visualization- IEEE JHAVE fosters. In this article author explains about the use of algorithm visualization as an effective learning platform for computer aided learner, which helps the student to better understand an algorithms. The Java-hosted algorithm visualization is a environment which support a variety of Algorithm visualization systems .It synchronizes its graphical displays using its engine with much effortless ways. i.e. Input generators , Information and pseudo code windows , stop-and-think questions, and meaningful content generation tools.[1] One of the existing AV engines supported by JHAVE is used for JHAVE-Hosted Automatic Visualization of Pointer operations. In the first step, by clicking on the button, the student starts up the JHAVEPOP client. then JHAVE server parses the input program, check its syntax errors and then sequence of snapshots are send back to the client. At this point, the first snapshot send by the server is shown in the main pane and the structured source code is shown in one of the side panes of JHAVEPOP window. Student can navigate these snapshots to verify that program is working properly orif not then find the reason. [2]

In this paper they studied the states of the field of Algorithm Visualization based on their analysis of collection of over

500 AV. They worked on what's needed in the AV, how they have been developed over time and are they really that helpful. From that they concluded that there is huge number of AV's but not high Quality visualization and more interaction of learners is still needed. Accordingly new AV should be designed and implemented so that it would give better results andmuch effective understanding and interactive learning tothe online learners.[3]

Towards developing an effective algorithm visualization tool for online learning (IEEE 2018). In that project they developed an effective AV tool that visualizes algorithms for online learners, improving their understanding about that algorithm. They mainly focused on pedagogy, usability and accessibility for online learners and accordingly designed for better user interaction and visualization and for this it involved 3 activities and various surveys for better designing i.e. establishingrequirements, designing alternatives and prototyping andevaluation. [4]

SYSTEM ARCHITECTURE

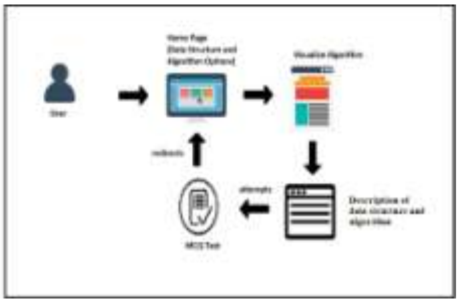


Figure 1. System Architecture

Created the website's User Interface (UI) using HTML, CSS. Implementation of animations, effects and core functionalities (data structures, sorting algorithms, path finding algorithms) using JavaScript (React Framework).

User will be directed to the homepage. In our homepage we have given various data structure & algorithm options that user can choose algorithm whichever user wants to learn.

Next user will move to the visualization page where user can see the visualization of the particular algorithm. Then user will visit the next page where we have given information of particular data structure or algorithm, where user can study and understand the algorithms theoretically.

Then user can attempt Assessments test where scores will also display from that user can understand how he/she understood about that particular algorithm. After that user will be directedto homepage.

RELATED WORK

Other notable systems include VPython and AlgoViz.org, which aimed to make algorithmvisualization accessible.

Web-Based Advancements (2000s):
With the rise of the internet, platforms like VisuAlgo and Sorting.at made algorithm visualization accessible online.
These platforms offered interactive visualizations of various algorithms and data structures, accompanied by detailed explanations.

Modern Systems (2010s-Present):
In recent years, platforms like DataCamp and LeetCode integrated algorithm visualization into their coding environments.

Open-source projects like Cmder and Visualgo.net gained popularity for their comprehensive collections of algorithm visualizations and educational resources.

METHODOLOGY

1) The objective of our project is to help learners understand the intricacies of algorithm and data structure implementations. The graphical and animation tools provide a step-by-step walkthrough of how operations are performed on the data structures andalgorithms.
2) Hence, creating a website for an effective learning of algorithms through visualization, we have first analyzed the

requirements of our project. We found out as much as algorithms for visualization and studied those algorithms in detail.

3) We found out the layout of our website as shown in the system architecture and according to that we worked.

4) At first, we have installed application like node.js in the visual studio code which comes with a bundle of (npm + npx) to run JavaScript in local system.

5) For creating the website's User Interface we used:

I. HTML, CSS: For designing and styling purpose.

II. Bootstrap: Used for further enhancing of thewebsite.

6) We have used various components of React.js and some of the libraries like react, react-Dom, react-move for the GUI logic, animation logic and Implementation part logic. JavaScript is used for the controlled execution of algorithms, implementation of animations, effects and core functionalities. Flex is used for array display and manipulation of positions.

Early Work:

Algorithm visualization began in the 1960s with efforts like Turtle Graphics for teaching programming concepts.
The 1980s saw Donald Knuth advocating for algorithmvisualization in his work "The Art of Computer Programming."

Pioneering Systems (1990s):

Systems like JHAVÉ and AVaJava emerged, providingvisualizations of algorithms and data structures.

7) We have used some of the new features like-

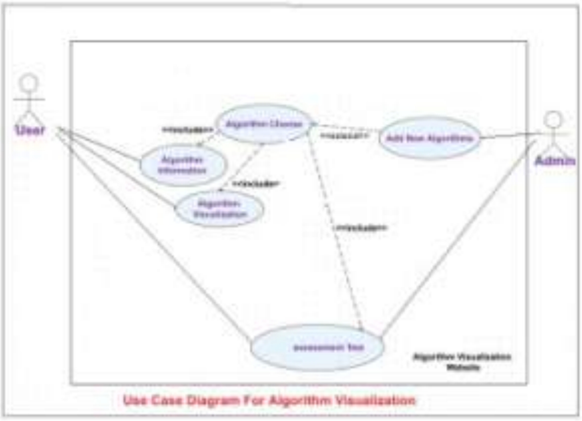
I. Speed of visualization: User can increase ordecrease the speed of visualization.

II. Play/Pause/Reset buttons: User can play, pause andreset the visualization specific Algorithm using this button.

III. Comparison of algorithms: In code or complexity of algorithm, usually we do not get that much of understanding of how much time is required to find the solution to that algorithm. Hence In order to decide which algorithm to choose over another and find out thetime required for the solution of problem we compare the algorithms visually.

IV. Information about algorithms is provided for user.

V. Assessment-Test: Using this assessments user can understand how much he is aware about the algorithms and how much he will understand the algorithms afterusing the website.



Use Case Diagram For Algorithm Visualization

IMPLEMENTATION

1. **Algorithm Visualization Components**: Describe the specific algorithms and data structures you've chosen to visualize on your platform. Explain how you implemented the visual representations of these algorithms, such as using graphical elements, animations, or interactive simulations.
2. **User Interface Design**: Discuss the principles of user interface design that guided the development of your platform. Explain how you created an intuitive and user-friendly interface to enhance the learning experience. Consider aspects such as layout, color scheme,typography, and navigation patterns.
3. **Frontend Technologies**: Detail the frontend technologies and frameworks you used to build the user interface of your platform. For example, discuss the role of HTML, CSS, JavaScript, and any frontend libraries or frameworks like React.js, Vue.js, or Angular.
4. **Backend Infrastructure**: Provide insights into the backend infrastructure supporting your platform. Explain how you manage user data, handle user authentication and authorization, and ensure the reliability and scalability of your application. Discuss the use of server-side

technologies like Node.js, Django, Flask, orothers.

5. **Data Management**: Describe how you handle data management tasks such as storing and retrieving algorithm visualizations, user progress, assessment results, and other relevant information. Discuss the choice of database systems (e.g., SQL, NoSQL) and data modelling techniques used in your application.
6. **Interactive Features**: Highlight any interactive features implemented in your platform to enhance user engagement and learning. This could include options for adjusting visualization speed, pausing/resuming simulations, toggling between differentalgorithm implementations, or exploring variations of algorithms.
7. **Responsive Design**: Explain how you ensured that your platform is accessible and functional across various devices and screen sizes. Discuss the use of responsive design techniques and media queries to adapt the layout and content based on the user's device.
8. **Performance Optimization**: Share insights into performance optimization strategies employed in your platform to ensure smooth user experience and fast loading times. This could include techniques such as code optimization, lazy loading of resources, caching, and minimizing network requests.
9. **Version Control and Collaboration**: Discuss your approach to version control and collaboration during the development process. Explain how you utilized tools like Git and GitHub for version control, issue tracking, and collaboration with team members.
10. **Testing and Quality Assurance**: Describe the testing methodologies and quality assurance practices employed to ensure the reliability, functionality, and usability of your platform. Discuss the types of testing conducted, such as unit testing, integration testing, user acceptancetesting, and usability testing.

ALGORITHM & TOOLS USED

- **Algorithms:**

1. **Sorting Algorithms**: Examples include Bubble Sort, Selection Sort, Insertion Sort, Merge Sort, Quick Sort, and Heap Sort.
2. **Searching Algorithms**: Common ones are Linear Search, Binary Search, Depth-First Search (DFS), Breadth-First Search (BFS), andA* Search.
3. **Graph Algorithms**: Include algorithms for traversing graphs, such as Depth-First Search (DFS), Breadth-First Search (BFS), Dijkstra's Algorithm, Bellman-Ford Algorithm, Floyd-Warshall Algorithm, and Prim's Algorithm.
4. **Tree Algorithms**: Algorithms like Binary Search Tree (BST) operations (insertion, deletion, search), Tree Traversal (Inorder, Preorder, Postorder), and AVL Tree operations.
5. **Dynamic Programming Algorithms**: Examples include Fibonacci Series, Knapsack Problem, Longest Common Subsequence (LCS), and Matrix Chain Multiplication.
6. **Pathfinding Algorithms**: Such as Dijkstra's Algorithm, A* Search Algorithm, and Depth- First Search (DFS) for maze solving.

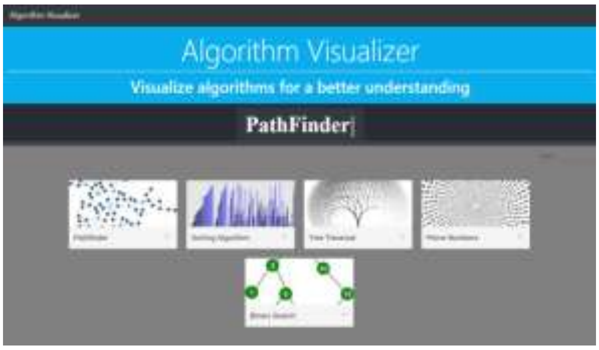
- **Tools:**

1. **JavaScript Libraries**: JavaScript is commonly used for implementing algorithm visualizations due to its compatibility with web browsers. Libraries like D3.js, p5.js, Three.js, and Phaser can be used for creating interactive visualizations.
2. **React.js**: A popular JavaScript library for building user interfaces. React.js can be used to create reusable UI components and manage thestate of the application efficiently.
3. **HTML and CSS**: HyperText Markup Language (HTML) is used for structuring the content of web pages, while Cascading Style Sheets (CSS) is used for styling and formattingthe visual presentation.
4. **Node.js**: A runtime environment for executing JavaScript code server-side. Node.js can be used to build the backend of web applications and handle tasks such as data storage, user authentication, and server-side processing.
5. **Bootstrap**: A front-end framework for developing responsive and mobile-first websites. Bootstrap provides pre-designed CSSand JavaScript components that can be easily integrated into web projects.
6. **Database Systems**: Depending on the requirements of your project, you may use relational database management systems (RDBMS) like MySQL, PostgreSQL, or SQLite, or NoSQL databases like MongoDB for storing and retrieving data.
7. **Version Control Systems**: Tools like Git and GitHub are essential for version control and collaboration among team members. They allow developers to track changes to the codebase, merge contributions, and resolve conflicts efficiently.
8. **IDEs and Text Editors**: Integrated Development Environments (IDEs) like Visual Studio Code, Atom, and Sublime Text, as wellas text editors like Vim and Emacs, are commonly used for writing and editing code.

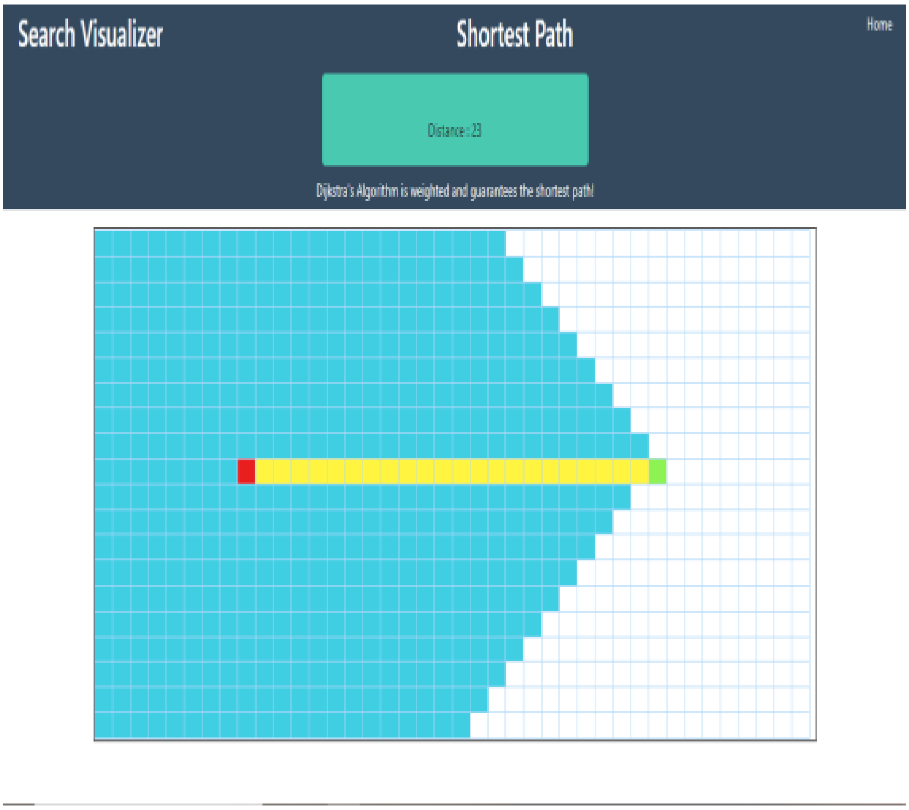
9. **Graphics and Animation Tools**: Software like Adobe Photoshop, Adobe Illustrator, and Sketch can be used for creating graphics and visual assets, while animation libraries like GreenSock Animation Platform (GSAP) can be used for animating elements on the web.
10. **Testing Frameworks**: Tools like Jest, Mocha, and Jasmine are used for writing and executing automated tests to ensure the reliability and correctness of the codebae.
11. Future Trends: Advances in AI, ML, and AR are expected to shape the future of algorithm visualization, offering more immersive learning experiences.

    Collaborative platforms and social features may play a larger role, fostering community-driven learning environments and knowledge sharing.

WEBSITE SNAPSHOTS



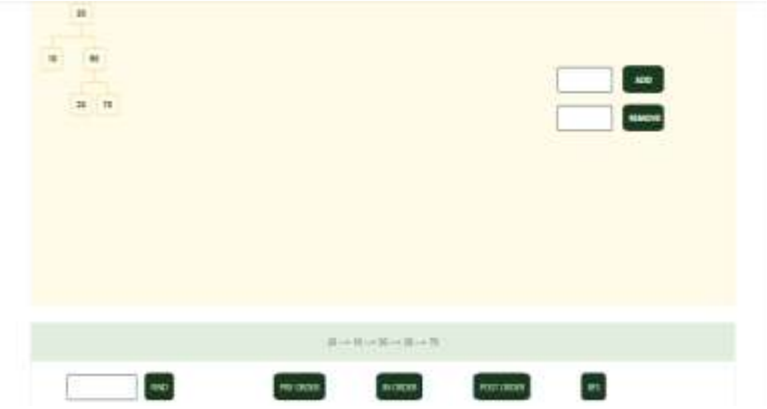
(a) Home Page



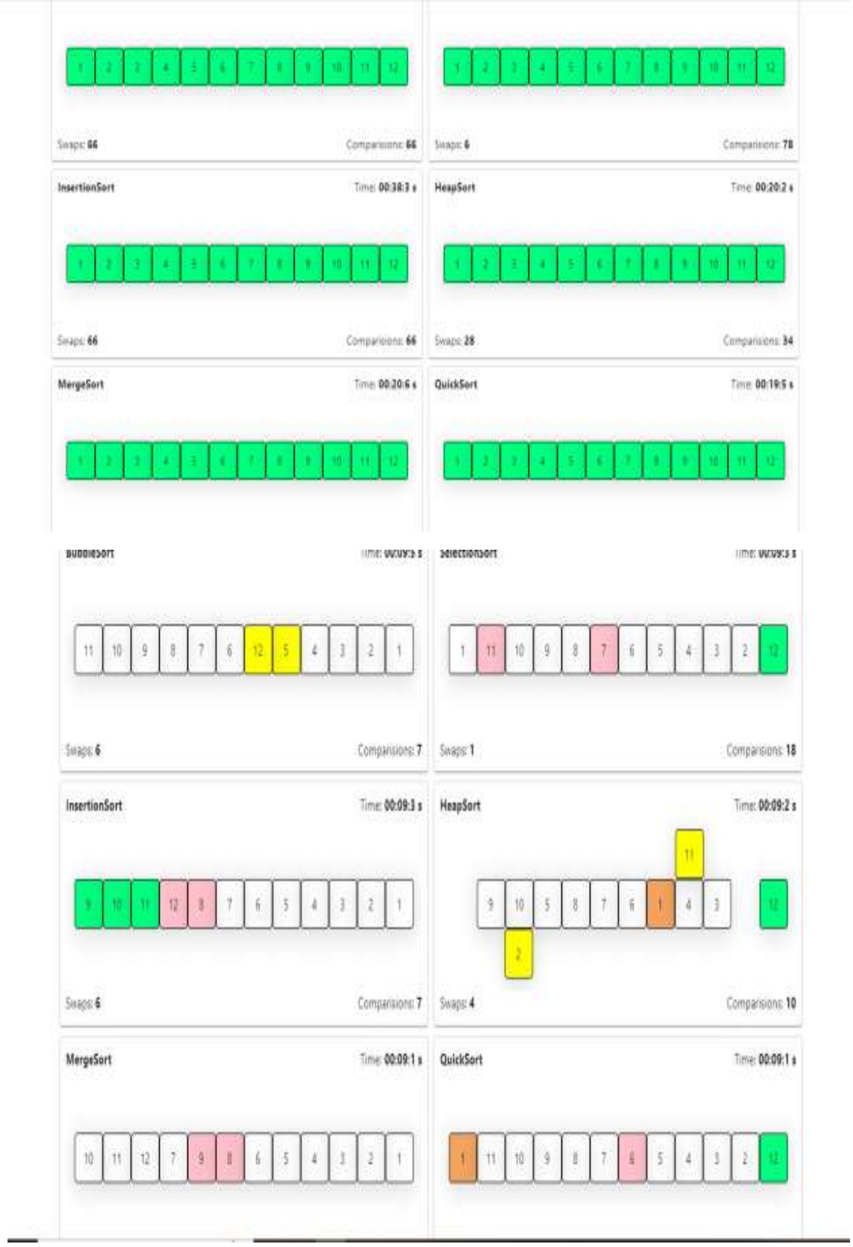
Distance : 23

Dijkstra's Algorithm is weighted and guarantees the shortest path!

b)Path Finder



(c) Tree Traversal



d)Sorting Visualization

CONCLUSION

Algorithm visualizer can be seen as a valuable supporting tool, used in various ways of education in the field of computer science. Our website helps to improve the quality of education in various field and it also contributes to the solution for some of the problems in higher education. Our intentions here is to provide better understanding of various algorithms like sorting algorithms and more complex data structures via their visualization. We will be providing some pre-assessment and post-assessment tests for more interaction of the learners. Thus, user can learn in much effective manner and also get their progress.

REFERENCES

[1]. Supporting algorithm visualization-IEEE JHAVE:https://ieeexplore.ieee.org/document/1510539

[2]. JHAVEPOP: visualizing linked-list operations in C++ and Java: https://www.researchgate.net/publication/262177625_J HAVEPOP_visualizing_linked-list_operations_in_C_and_Java

[3]. Developing an effective algorithm visualization tool for online learning: https://www.researchgate.net/publication/329481179_ Towards_Developing_an_Effective_Algorithm_Visualization_Tool_for_Online_Learning

[4]. Developing an effective algorithm visualization tool for online learning (IEEE 2018).https://www.cs.utep.edu/makbar/papers/TOCE.p df

[5]. Course referred for development of the project: https://www.udemy.com/course/the-complete-web-development- bootcamp/learn/lecture/12299730?start=120#content

[6]. Research paper study Algorithm Visualization: https://www.cs.utep.edu/makbar/papers/TOCE.pdf

[7]. Explorin the role of visualization and engagement in computer science education: https://www.academia.edu/75893135/Exploring_the_ro le_of_visualization_and_engagement_in_computer_science_educationn

[8]. A Survey of Successful Evaluations of ProgramVisualization and Algorithm Animation

[9]. Systems https://www.academia.edu/76408518/A_Survey_of_Successful_Evaluations_of_Program_Visualization_and_Algorithm_Animation_Systems

[10]. Learning Bootstrap5 for project front-end :https://getbootstrap.com/docs/5.0/getting- started/introduction/

[11]. Referring Html and CSS: https://www.w3schools.com/html/html_css.asp