iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** II **Month of publication:** February 2024

DOI: https://doi.org/10.22214/ijraset.2024.58613

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue II Feb 2024- Available at www.ijraset.com

Visualizing Algorithms in the Field of Education: A Comprehensive Review

Varun Singh[1], Sudhanshu Dubey[2], Asst Prof. Er. Asim Ahmad[3]
Computer Science and Engineering SRMCEM Lucknow India

Abstract: ***Visualization of algorithms can enhance computer science education by offering a beneficial method for comprehending intricate concepts within engineering courses. An electronic learning tool has been created to cover various algorithms, including sorting algorithms, prime numbers, recursive sorting algorithms, and the N Queen problem. Employing animation as a means of conveying information, the tool enables users to select data ordering and algorithms. Users can either initiate automatic animation or progress through the steps at their own pace. Additionally, the pathfinding algorithm permits users to modify the starting and ending nodes. This e-learning tool serves as both a standalone application and a complementary resource for in-person training, providing a visual representation of the execution stages of algorithmic rules.***
Keywords: Data Structure and Algorithm, Algorithm Visualization, Sorting Algorithms, Pathfinder, Prime Numbers, N-Queen Problem

I. INTRODUCTION

The Algorithm Visualizer is an interactive website that uses code to display various algorithms' visual representations. The method of visualizing an algorithm greatly simplifies understanding of the procedure.

The main goal of the Algorithm Visualizer is to establish a platform facilitating individuals' comprehension of the functioning of various algorithms. The website designers made a concerted effort to provide users with the best possible experience by creating the demonstration software in an easily understandable and user-friendly format[1]. Users are encouraged to enhance their learning experience by experimenting with each algorithm on their own dataset.

This online educational resource presents the fundamental principles of widely recognized algorithms, demonstrated through computer programs. It serves as a valuable learning resource for students aiming to acquire a deeper understanding of algorithms. Also, the website might be easily used as a demonstration by lecturers and tutors during classes[1]. To enhance comprehension, visitors to this website are recommended to have a foundation in programming and be acquainted with basic data structures like arrays, lists, and trees, as well as an understanding of recursive processes. Moreover, having a grasp of fundamental algorithms and their applications would be beneficial.

Algorithm visualization is a type of software visualization that uses dynamic graphical representations to depict the operation of an algorithm or program [2]. A taxonomy of visualization methods is presented in [3], along with a review of factors that enhance the efficacy of software visualization in [4]. Our tool, Algorithm Visualizer, can function as a standalone; however, its advantages are more significant when utilized as support for a data structures and algorithms course[5]. With the help of Algorithm Visualizer, students may take charge of their own education and strengthen their grasp of foundational material, giving their lecturer more time to deliver more challenging and engaging material and pique students' interest and enthusiasm to learn.

II. LITERATURE REVIEW

Because it makes studying algorithms more visual, the interactive platform Algorithm Visualizer has drawn a lot of interest in the field of computer science education. Numerous investigations have been carried out to evaluate its possible uses and efficacy. Below are some instances of pertinent literature:

1) The BALSA (Brown University Algorithm Simulator and Animator) was developed in the 1980s to help students understand computer algorithms by visualizing their performance [6]. It offered users control over display features, multiple algorithm views, and interaction with the animation. The program saved the algorithm's history and user window settings. BALSA was written in C and used black and white animation.[7]

2) XTango, a color-enhanced, 2.5D animation system for algorithms and programs, uses the transformation paradigm to achieve smooth animation. It is distributed with sample animation programs and can be embedded in C or other programming languages.[7]

©IJRASET: All Rights are Reserved

3) DynaLab was created in the 1990s at Montana State University to provide an interactive and visual laboratory in computer science. It includes X-windows and MS- windows program animators, an animation program library, and the ability to undo animation. DynaLab was intended for use in laboratory settings to perform tests with algorithms.[8] iv. SWAN, developed at Virginia Tech, visualizes C/C++ data structures as graphs. It has three main components: SAIL, SVI, and the Swan Kernel. SAIL creates system views, SVI tests the annotation system, and the Swan Kernel is the main module.[9]

III. PROPOSED METHODOLGY

This is a summary of the suggested method for creating a framework that shows how an algorithm works:

1) A user-friendly web interface that enables users to select their preferred algorithm.
2) Upon selecting the algorithm, users will input their own data/values, or they can opt for random values.
3) The algorithm's operation will be displayed via dynamic animations, aiding users in comprehending its working principles.

IV. MODULE DESCRIPTION

The visualization module explains algorithms and data structures with interactive graphics. Its goal is to enhance algorithm instruction and learning by:

1) Using dynamic visualizations in place of static ones.
2) Encouraging pupils to utilize their own examples, as this can improve their comprehension of algorithms.
3) Improving comprehension of algorithmic operations, which may allow educators to concentrate more on sophisticated applications.
4) Giving pupils a single platform for all of their learning needs by providing a uniform interface for visualizing various algorithms.

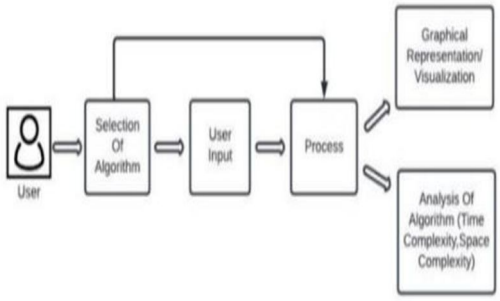


Figure 1: Block Diagram of Model

a) LAYER 1:

This layer gives the user the option to choose the visualization algorithm of their choice. With interesting animations, a learner can quickly understand how a sorting algorithm works, as seen in the image.

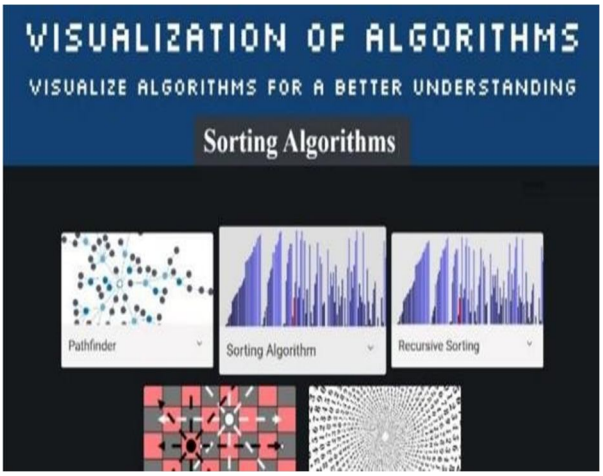


Figure 2: Home Page of Website

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue II Feb 2024- Available at www.ijraset.com



Figure 3: Selection of an Algorithm

b) LAYER 2:

User input is a part of this project's second layer. The user has two choices after choosing the algorithm they wish to study:

i. they can either enter the values themselves.

ii. Random values may be used.



Figure 4: Sorting Random Values

c) LAYER 3:

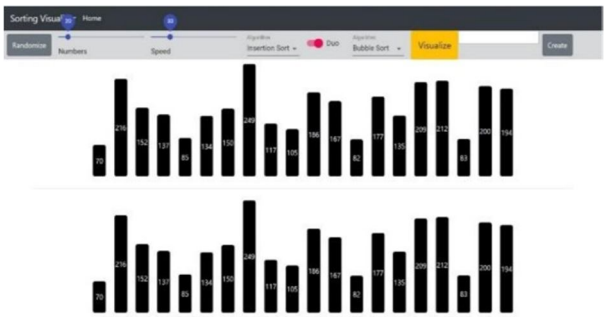
The algorithm visualization process starts once the user enters the values. Using the insertion sort example from above, the learner can interact with and see the algorithm in action by opening a new window that displays the exact array from the question.



Figure 5: Insertion Sort Visualization

d) LAYER 4:

Users can compare the sorting speeds of each algorithm by using the Algorithm Visualizer website's component that lets them compare two algorithms at once through visualization. By offering a side-by-side comparison, this feature improves the learning process by assisting users in comprehending the advantages and disadvantages of each algorithm.


Figure 6: Comparison of Insertion Sort and Bubble Sort

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue II Feb 2024- Available at www.ijraset.com

V. IMPLEMENTATION

Algorithm Visualizer is a initiative providing an interactive platform designed for instructing and grasping algorithms via simulation. The system is implemented using just three programming languages: HTML5, CSS, and React JS, enhancing efficiency and user-friendliness for learners. HTML5 and CSS are responsible for constructing and maintaining the user interface, while React JS executes the algorithms and dynamically updates the interface. The communication between HTML5 and React JS is bidirectional, ensuring smooth integration and real-time updates of the interface. The adjusted React JS code adopts a more object-oriented approach, enabling improved function calls for each button.

The Algorithm Visualizer system's back end interface is limited to selecting algorithms and navigating via buttons to play the animation. Because of this simplification, students can choose the algorithm they want to see and follow along with ease. The system also includes a time complexity parameter, which indicates how long a sorting algorithm takes to run.

This feature is helpful since it lets students evaluate and contrast the effectiveness of various methods.

The Algorithm Visualizer project's primary goal is to give students access to an interactive platform for studying and teaching algorithms. The system's sophisticated algorithms, intuitive UI, and time complexity settings can improve students' comprehension of computer science ideas. The system is accessible to students of all ages due to its utilization of HTML5, CSS, and React JS.

VI. RESULT

The n-queen problem is one of the many algorithms that students can interactively examine on the Algorithm Visualizer website. A classic computer science puzzle called the "n-queen problem" consists of arranging n chess queens on an n×n chessboard in such a way that no two queens can threaten to topple the other.

Through the Algorithm Visualizer website, students can see how the n-queen issue is solved. With the help of the system, students can choose the chessboard's dimensions and the quantity of queens to be placed on it.

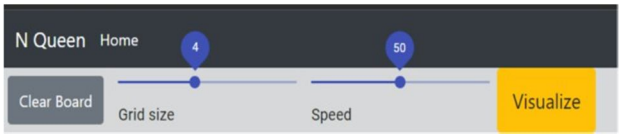


Figure 7: Selection of Number of Grids and Speed of Visualization

After the parameters are established, students can execute the algorithm and see how the queens are arranged on the board to meet the limitations of the task.

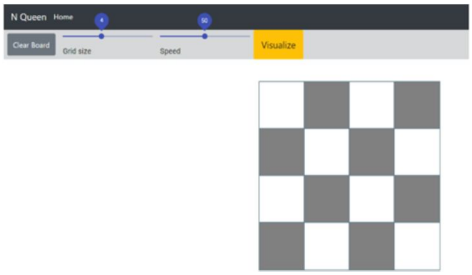


Figure 8: Showing 4x4 Grids

The website shows the current condition of the board, with the queens positioned in their current locations, as the algorithm runs.

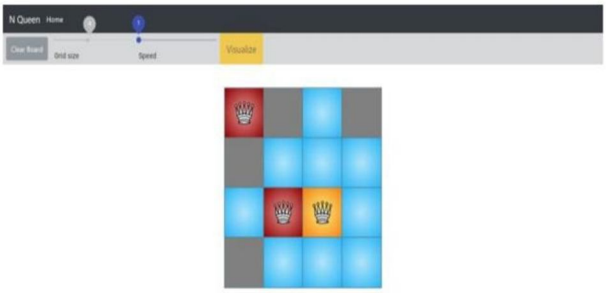


Figure 9: Placing queens at different grids.

©IJRASET: All Rights are Reserved

The algorithm returns to the prior state and looks for an alternative solution if a conflict is identified.
Learners can get a better knowledge of backtracking algorithms and how to apply them to tackle challenging issues by utilizing the n-queen problem as an example.
Because the Algorithm Visualizer website is interactive, students can experiment with various settings and see how they impact the problem's solution.
In conclusion, backtracking algorithms can be used to solve the n-queen problem, a well-known computer science puzzle.

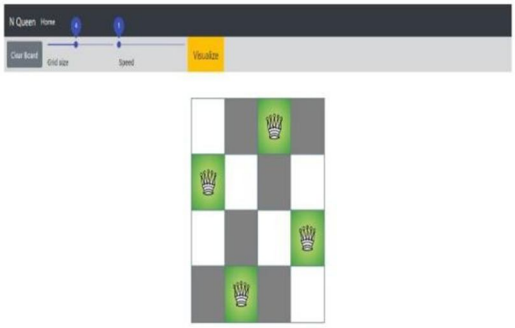


Figure 10: N-Queen Algorithm Visualization

Students can use the Algorithm Visualizer website to see how the n-queen issue is solved, play about with various settings, and see how those changes impact the final result.
With the use of this program, students can gain a deeper comprehension of backtracking algorithms and how they are used to solve challenging situations.

VII. CONCLUSION & FUTURE SCOPE

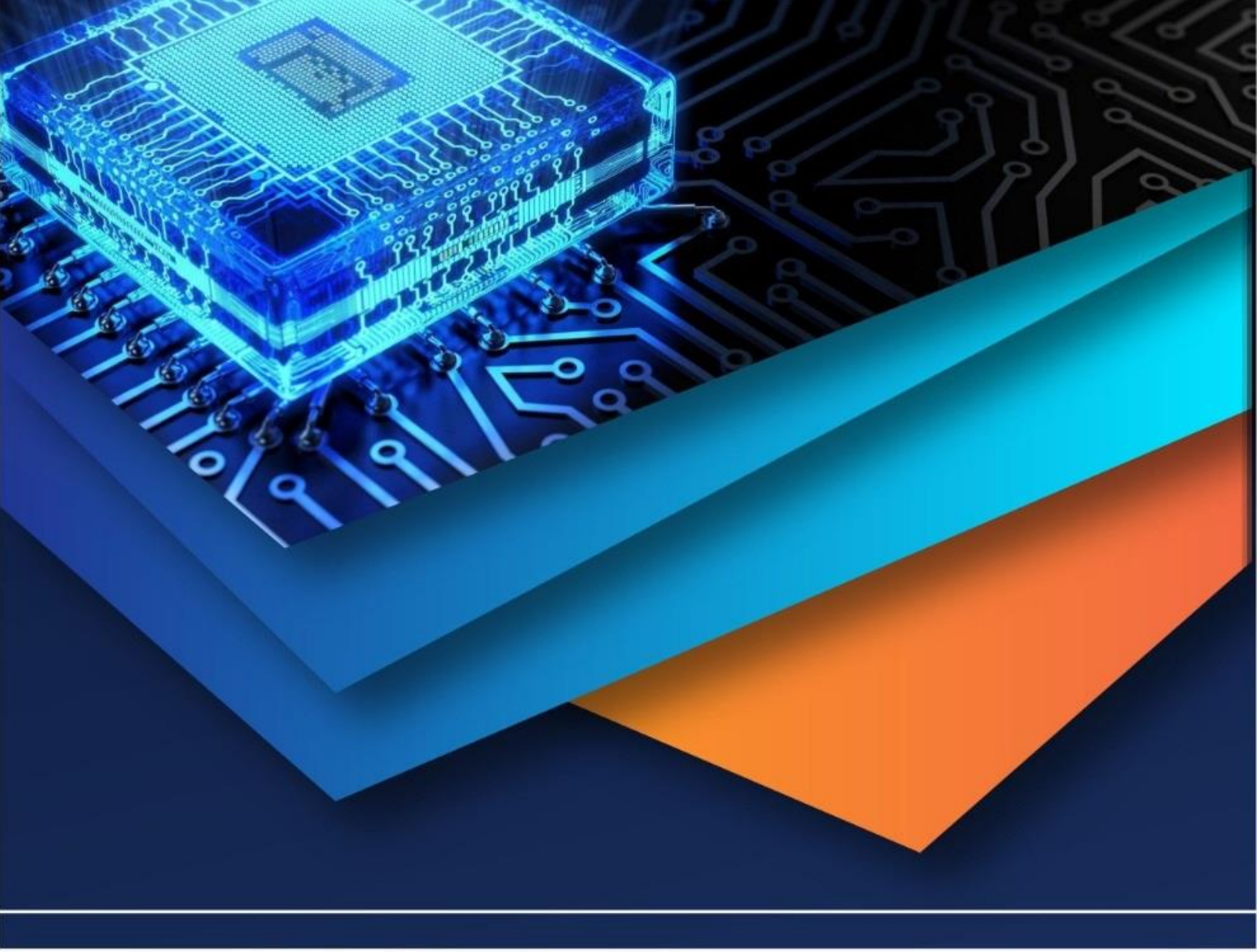
After doing some study, we discovered that there were no appropriate ways for professors to successfully and simply communicate their concepts to students. We created an application with several advantages to solve this problem.
First, we discovered that visual learning helps people remember topics more effectively than verbal or written explanations. For this reason, our program has interactive visuals to improve the educational process.
Second, to keep users of all ages interested, our program is easy to use and offers a variety of entertaining tasks like visualizing through mazes and patterns.
Furthermore, the algorithm visualization app has almost all prominent and well-known algorithms, including sorting and pathfinding algorithms, making it a thorough learning resource for students with an interest in this subject.
By including the suggested elements, our algorithm enhances students' understanding and memory of ideas while also facilitating a smooth transition from secondary to tertiary education, so increasing their productivity and competency.
In conclusion, by encouraging a deeper knowledge of algorithms through interactive visualizations, the Algorithm Visualizer project has the potential to greatly enhance computer science education.In conclusion, by encouraging a deeper knowledge of algorithms through interactive visualizations, the Algorithm Visualizer project has the potential to greatly.
There is potential for Algorithm Visualizer to be developed further in the future. Adding more sophisticated algorithms and data structures to the mix of visualization-ready algorithms is one possible path. Using machine learning techniques to generate customized learning experiences for each student is another area that needs work.
Incorporating analytics tools and feedback mechanisms can also yield important insights into how students learn and how the platform can be further enhanced. Algorithm Visualizer has the potential to be an even more useful teaching tool for computer science with these improvements.

REFERENCES

[1] “Algorithm Visualizer” by Roshni Gurubaxani, Virat Kuma, Priyanka Kumari, Tejanki Ambrale, Prof. Dhanashri Nevase
[2] M.E. Tudoreanu, R. Wu, A. Hamilton-Taylor, and E. Kraemer, "Empirical Evidence that Algorithm Animation Promotes Understanding of Distributed Algorithms," in Proceedings of the IEEE Symposium on Human Centric Computing Languages and Environments, HCC02, held in Arlington, Virginia in September 2002.
[3] B.A. Price, R.M. Baecker, and I.S. Small, "A Principled Taxonomy of Software Visualization," Journal of Visual Languages and Computing, vol. 4, no. 3, pp. 211-266, 1993.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

[4] Ž. Šuchová, "Visualization of Algorithms and Data Structures," Bachelor thesis, DCI FEEI TU of Košice, 2010 (in Slovak).
[5] Erin Teo Yi Ling(2014). “Teaching Algorithms with Web-based Technologies.”
[6] Shaffer, Clifford & Cooper, Matthew & Alon, Alexander & Akbar, Monika & Stewart, Michael & Ponce, Sean & Edwards, Stephen. (2010). Algorithm Visualization: The State of the Field. TOCE. 10. 10.1145/1821996.1821997.
[7] “A Literature Review on Algorithm Visualizers” Sweeta Bansal, Karan Kohli, Krishna Kumar Vishwakarma , Kush Gupta, Inderprastha Engineering College 2022
[8] Michael R. Birch, Christopher M. Boroni, Frances W. Goosey,Samuel D. Patton, David K. Poole, Craig M. Pratt, Rockford J. Ross. “DYNALAB A Dynamic Computer Science Laboratory Infrastructure Featuring Program Animation”
[9] Shaffer, Clifford & Heath, Lenwood & Yang, Jun. (1996). “Using the Swan data structure visualization system for computer science education. ACM Sigcse Bulletin.”
[10] “E-learning Tool for Visualization of Shortest Paths Algorithms” by Daniela Borissova and Ivan Mustakerov, ResearchGate, July 2015.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)