IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Algorithms Visualizer application

Aditya[1], Shipra Srivastava[2],Gulshan Gupta[3], Bilal Ibrahim[4], Jatin Kumar[5]

[2]Assistant Professor of Information Technology Engineering, Greater Noida Institute of Technology

[1,3,4,5]Student, Department of Information Technology Engineering, Greater Noida Institute of Technology

Abstract – Algorithm visualization has been high topic in CS education for years, but it did not make its way to university lecture halls as the main educational tool. The present paper identifies two key conditions that an algorithm visualization must satisfy to be successful: general availability of used software, and visualization of why an algorithm solves the problem rather than what it is doing. One possible method of "why" algorithm visualization is using algorithm invariants rather than showing the data transformations only. Invariants are known in Program Correctness Theory and Software Verification and many researchers believe that knowledge of invariants is essentially equivalent to understanding the algorithm. Algorithm invariant visualizing leads to codes that are computationally very demanding, and powerful software tools require downloading/installing compilers and/or runtime machines, which limits the scope of users. One our important finding is that, due to computing power of the recent hardware, even very complex visualization involving 3D animation (e.g., Fortune's algorithm, see Section 4) could be successfully implemented using interpreted graphic script languages like JavaScript that are available to every web user without any downloading/installation.

Key Words: Data Structure and Algorithm Visualizations, Algorithm Animation, E-Learning.

1. INTRODUCTION

Algorithm visualization (often called algorithm animation) uses dynamic graphics to visualize computation of a given algorithm. First attempts to animate algorithms date to mid-80's (Brown, 1988; Brown and Sedgewick, 1985), and the golden age of algorithm visualization was around the year 2000, when excellent software tools for a dynamic algorithm visualization (e.g., the language python and its graphic libraries) and sufficiently powerful hardware were already available. It was expected that algorithm visualization would dramatically change the way algorithms are taught. Many algorithm animations had appeared, mostly for simple problems like basic tree data structures and sorting. There were even attempts to automatize development of animated algorithms and algorithm visualization. Another direction was to develop tools that would allow students to prepare their own animations easily. Instead of giving particular references to algorithm animation papers, the reader is directed to a super-reference (Algo viz,) that brings a list of more than 700 authors, some of them even with 29 references in algorithm animation and visualization. There are also many webs. Pages that offer algorithm animation systems, e.g... (Algoanim, Algoma tion,; DD2,; Algo Liang, ; Visual Algo,).

However, algorithm visualization and animation has not fulfilled the hopes, and it is still not used too much in CS courses. One can even find articles with titles like "We work so hard and they don't use it" (basset Levy and Ben-Ari, 2007), complaining about low acceptance of algorithm animation. Tools by teachers. The number of articles, reports, and visualization tools sensibly declined in the second decade of the new millennium. The present paper is an attempt to find why algorithm animation and visualization is used much less in instruction then we hoped 10 or 20 years ago. We strongly believe that the reason is relatively simple: An algorithm operates on some data (the input data, working variables, and the output data). Usually, in any particular field of Computer Science, there is a standard way of visualization of data - graphs and trees are drawn as circles connected by line segments, number sequences could be visualized as collections of vertical bars, there are standard ways of drawing matrices, vectors, real functions, etc.

An algorithm animation is usually implemented by running the algorithm slowly or in steps, and simply modifying the visual representation of the data in the screen. A person who knows and understands the algorithm in question can see how the algorithm progresses, but a novice user just see visual objects moving and changing their shapes and colors, but finding out why the movie runs in that way is usually too difficult for him or her.

The solution that we offer is to visualize (not as much) what the algorithm is doing, but why it is working in the way it is working. In other words, our aim is to visualize an abstract algorithmic idea that is behind a particular computing method. We admit that the statement is rather vague. Moreover, we are not able to give any general methodology of visualizing abstract algorithmic ideas (and we guess that no such methodology exists). Nevertheless, certain examples are given in an attempt to illustrate the approach. As we argue below, understanding an algorithm is essentially equivalent to the knowledge of the invariant used to prove its partial correctness and termination. Since the notion of an algorithm invariant is less abstract and vague than the term "algorithmic idea", we believe that "animation of algorithm invariants" is a better description of our work.

The purpose of this paper is to provide a summary of preliminary findings resulting from our efforts to survey the state of the field of algorithm visualization. To bound the content area, we focused our attention on topics commonly taught in undergraduate courses on data structures and algorithms. We seek an understanding of the overall health of the field, and present a number of open research questions that we and others can work on in the future. Some examples of the questions we seek to address are:

- What visualizations are available?
- What is their general quality?
- Is there adequate coverage of the major topic areas covered in data structures and algorithms courses?
- How do educators find effective visualizations?
- Is the field active, and improving?
- Is there adequate infrastructure for storing and disseminating visualizations?

1.1 Objective

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. And obviously it is needless to say that anyone who is willing to contribute is in voted to use their creativity in making the visualizations even better and attractive. One can add fresh Algorithms and visualization of their choice too.

- This project is for educational purpose.
- Extract the required feature and classify among others.
- Users able to establish a relation between a property and a table simply by clicking on a row and dragging it over the target table. Tables are rendered in HTML inside of a div. An SVG HTML element is placed behind the tables' div. This is where we'll draw the relations. When table positions change, our lines need to be redrawn.

1.2 PROPOSED METHODOLOGY

As mentioned above, we already have an existing application with existing functionality of basic algorithm visualization of both path-finding and sorting algorithms, mazes and patterns and displaying time complexity [1]. Now, to make the visualizations all the more realistic and easier to comprehend we decided to include a few additional features in our application. The additional features to be included are as follows:

• CPU Scheduling algorithms –

There can be a number of processes running parallel in a system. So, the CPU (Central Processing Unit) must come up with a way by which it can choose from the waiting processes and complete their execution one by one. In doing so CPU must ensure the following things:

a) Maximum resource utilization
b) Minimum response time
c) CPU allocation must be fair enough
d) Maximum throughput

Considering the above criteria, the various popular scheduling algorithms are as follows:

a) First-Come, First-Served (FCFS) Scheduling
b) Shortest-Job-Next (SJN) Scheduling
c) Priority Scheduling
d) Shortest Remaining Time
e) Round Robin (RR) Scheduling
f) Multiple-Level Queues Scheduling

Imagining and understanding something which is not physically visible to us, can be difficult at times, especially for students. Visualization of the processes on the basis of their overall time required for CPU execution (i.e., Burst time) and selection of various processes on the basis of the algorithm one by one, can be used to provide a clear understanding of how the algorithm is working in the actual scenario. Thus, we can add this feature of scheduling algorithm visualization and other topics to expand the knowledge base of our application.

• Tree based visualization of algorithms

We have observed that in most of the standard books for algorithms, the working of algorithms is shown with reference to tree data structure. In order to ensure a gradual shift from books or classroom technique to e-learning, we can introduce this feature in our application and then in advanced levels we can introduce the existing, more efficient form of algorithm visualization for the students

• Visualization assisted by line-by-line code execution

Usually, the algorithms are given in the form of pseudo code or step wise procedure in most of the books and websites. Also, students are expected to write the

working of algorithms in the same manner in their exams. So, in order to ensure there is no gap between the understanding and presentation of the concept in exams for the students, we can inculcate a feature in the application which can help students understand the working through step-wise visualization of a particular algorithm. One possible way of implementing this feature is to divide the visible area of the screen into two parts, one part displaying the visualization and another representing the algorithm, where line of code corresponding to the current portion of visualization is highlighted, thus enhancing students' retention capability and presentation skills in the domain.

• Pause and Play

All of us must be familiar with the condition while watching a video, we have to pause the video due to some work or urgency or while talking to someone when there is an interruption, we ask the person to stop and then continue later when we have dealt with the interruption. In such situations, we do not start watching the video from the starting or we do not ask the other person to start all over again. Similarly, we do not want our users to suffer due to interruptions. Thus, we can introduce a pause feature, which would enable the user to temporarily stop the visualization and later resume from where he/she had left the visualization. In case the user wants to visualize again from the starting, that option would always be available for the users irrespective of the current state of application.

• Real world application simulations

We have tried to provide the best possible form of visualization through our application, but scope of improvement is always there. So, in order to provide the students with a view of how these algorithms would be applicable in real world scenarios, we can have various simulations as follows:

a) Real time geo-location and navigation in maps, Maps and route trackers have made our lives and travelling a piece of cake. Thus, demonstrating how shortest route-finding algorithms are being implemented in maps and other services might suffice the students' thirst of curiosity.

b) Delivery and logistics management On-time delivery has
turned out to be one of the most crucial aspects of the existing e-commerce applications to retain and satisfy their customers in times where there is cut-throat competition. The ecommerce industry needs to find our best possible solutions to reach their customers in time and ensure optimum resource utilization at the same time. Thus, not only the students but also various other industries can use this application to find out solutions to their problems.

PROPOSED SYSTEM

Algo Assist is an integrated platform where students can enhance their coding skills, teachers can evaluate student's work and the focus is on 'Algorithm Visualization' for a better understanding of the algorithm's flow and operations. It contains lab integration into one platform which makes it more feasible for teachers and students to use. This platform consists of three user groups namely student, teacher, and developer.

Figure 1 depicts the system architecture. The client-side and server-side are connected using REST API, which performs retrieve and CRUD operations on the server. The server contains server-side and client-side scripts and is connected to the MongoDB database.

When student's login into the platform, they will land upon a fully interactive dashboard. This dashboard will give them their performance metrics, leaderboard, personal profile, assignment, discussion forums, and subjects. Each subject will be organized into pre- and post-assessments, brief textual explanation, line-by-line visualization, and coding playground. First, pre-assessment will be taken which will be beneficial for students to understand their prior knowledge related to the selected algorithm/topic. Then a brief textual explanation of the algorithm will be given so that students should have some prior knowledge before directly going for the visualization. To understand algorithms better, being familiar with the significance of each line, understanding how each line is responsible for manipulating data, what changes a line makes in the input data provided, which line is the crux of the algorithm, and from where and in which direction the data flows play a very crucial role in understanding the subject better.

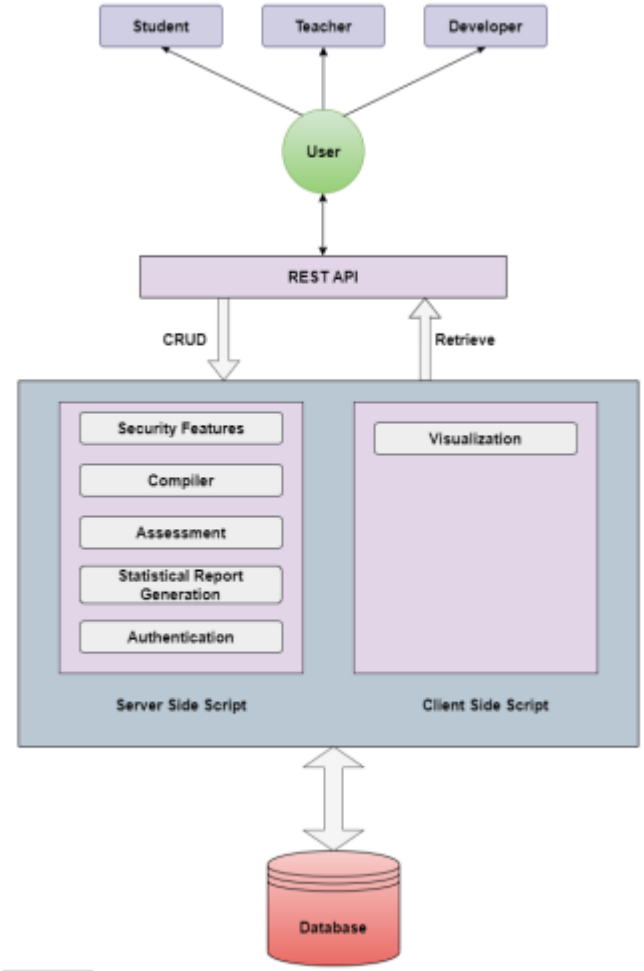

Fig.1. Proposed ER model of Algorithm Visualizer

This is where the visualization comes into the picture. Visualization of algorithms will help students understand the algorithms in a precise and accurate manner. Further coding playground will help students to get an understanding of the code and implement the algorithm programmatically. The coding platform will have compiler/interpreter support of various languages. Post assessment will finally test their overall learning from the module. The scores of the assessments will be used to generate leaderboards standings, personal growth metrics, and provide feedback for improvement (if needed). In the assignment, students will have to submit the assignments before the due date given by teachers. In the discussion forum, students can post their difficulties. Their academic-related doubts will be solved by teachers or other classmates and their system-related doubts will be solved by developers. When teacher's login into the platform, over the dashboard they will see a leaderboard, assignment allocation, assignment evaluation, student's report, and discussion forum. Inside the leaderboard, they can see the progress of students in a class. In a student's report, they can see overall class performance and modules completed by each student. In the assignment allocation section, teachers can assign various assignments or organize a quiz for students, set the due date, and send notifications. Assignment evaluation can be done manually or automatically. They can also post their doubts related to the system in the discussion forum. Another main task of teachers is to add students to the system and provide the credentials. The developer group holds supreme power over the application. They maintain the whole system. They can create a visualization of new algorithms, add or edit assignments and can also assign roles to the users. They can register the teacher on the platform and provide the credentials. For security, the application will have copy/paste disabled. During the test, switching tabs will cause the test to end. The test will be proctored by webcam and it will have an inbuilt timer for the test.

2.1 MODELING AND ANALYSIS

According to the proposed features the improvised model of the application will be as follows:

The basic architecture of algorithm will remain the same as earlier which included basic visualizations and mazes and patterns. To incorporate the new feature the following would need to be done:

a) Addition of CPU scheduling algorithms to the existing set of algorithms for visualization

b) Pause and Play buttons to allow user the user to temporarily stop and play the visualization as and when required

c) Asking the user to select from the available types of visualizations they prefer as now the user would have a choice for tree-based visualization also.

d) Allow the user to view the pseudo code for the currently selected algorithm alongside so the user can relate the working of the algorithm with the corresponding steps.

e) Give the user options to relate the algorithms with real world simulation environments.

RESULTS AND DISCUSSION

• We conducted a survey to find out if there is any considerable impact in the people willing to choose the Algorithm Visualizer application with the proposed features. To our delight we found 10% increase in the overall survey result, that is, now 70% people are willing to refer to algorithm visualizer application with improved features for understanding algorithms as compared to 60% of last time

.• It has been found by researchers and scholars that when people try to memorize something in the form of a storyline or relating that aspect with daily lives then they are more

likely to retain it for long period of time. Thus, the additional features will make the students retail the concepts for long and guide them to becoming specialists in this domain.

• Moreover, relating the concepts learnt in course curriculums to real life examples will develop a research perspective in students and knowing the actual working of anything around them will turn out to be their habit, thus, contributing to their knowledge.

• This improved application would enable the teachers to explain difficult topics with ease thus, helping them get off the burden. Also, the teachers themselves can come up with innovative ways of visualizations that can help improve the application's performance from time to time.

DESIGN CONSIDERATION

A. System Features

Visualization of algorithms: Learning algorithms through visualizations can help students grasp algorithms better. Having a good and basic knowledge of DSA is important to become a good developer. Algo Assist ensures to provide a clear visualization and help students understand algorithms.

Connecting students and teachers: Algo Assist aims to connect students and teachers in one place. Teachers can assign tasks and track the progress of students. Students can effectively learn algorithms and data structures. Algo Assist aims to connect and enhance the learning experience.

Security features to avoid cheating: Algo Assist is designed keeping in mind that it will devoid students from using malpractices such as copying code from online resources and copy-pasting from the local clipboard. This will be achieved by disabling right clicks, disallowing tab switches, and enabling full-screen mode.

Open for improvement: Any person with the developer's privileges can add visualizations and content to the application. This feature keeps it open for adding algorithm visualizations and new features to the application.

B. Assumptions and Dependencies

- Teachers' and students' access control is predetermined. Developers will add teachers to the application and teachers will add students and provide them login credentials
- . Access control and authorization will be handled by REST API.

C. External Interface Requirements

User Interface: The user interface will provide a good look and effects to make it more user-friendly. Teachers and students can operate the system very efficiently.

Communication Interfaces: The communication in the system would include email services, web server communications, etc. HTTP would be the communication standard used. The frontend application will be communicating with the REST API, which performs the CRUD operations on the database.

D. Operating Environment

The application will have ReactJs as frontend and Django as backend with MongoDB as a database. The frontend of the application i.e Reactjs will interact via REST API with the server in the backend which runs Django. Compilers and interpreters for different programming languages will be integrated with the platform.

E. Functional Requirement

- Student Activities: These activities include giving assessments, understanding algorithms through visualization, submitting assignments, coding and checking the leaderboard.
- Teacher Activities: Teacher activities include view students' performance, giving assignments to students, assignment evaluation and enrollment of students,
- Developer Activities: These activities include adding algorithm visualization, management of all the legalities,

verification, enrollment of teachers, and addition of new features.

4. CONCLUSIONS AND FUTURE PLANS

In a nutshell, we identify some issues by experiencing them ourselves in the present learning strategies in use and we tried to help better the scenario for aspiring students in this domain through or progressive web application. When we ourselves were learning the subject of algorithms in our curriculum, we found it a bit difficult to relate and understand the practical implementation of the algorithms owing to the difficulty in communication of the concepts from the teachers to the students. We found that there were no proper means that the teachers could adopt to portray their ideas in a better and easy manner in front of the students. So, we built an application which could help in the following ways: -

• It has been found that it becomes easier for humans to retain the concepts when learnt through visuals than just textual or speech explanations.

• Application is extremely user friendly so people of any age can engage and start learning new things right away. The application would also include various fun filled activities like visualization through mazes and patterns.

• This application will also include a parameter of time complexity which will be displayed after the particular sorting algorithm has completed its execution for better comparison.

• Almost all the famous and important algorithms will be present in the application for visualization with both path-finding and sorting algorithms present in same application, thus making it a one stop destination for the students of this domain

• With the inclusion of the proposed features, the algorithm would not only help in better visualization and retention of concepts by students, but also, enable the students have a gradual and smooth transition from school level to college level of education, thus, enhancing their knowledge and productivity.

REFERENCES

[1] “Algorithm Visualizer” by Barnini Goswami, Anushka Dhar, Akash Gupta, Antriksh Gupta, Volume 3 Issue 03 March 2021, International Research Journal of Mordernization in Engineering Technology and Science.

[2] “E-learning Tool for Visualization of Shortest Paths Algorithms” by Daniela Borissova and Ivan Mustakerov, ResearchGate, July 2015.

[3] “Algorithm Visualization: The State” of the Field by Clifford A. Shaffer, Matthew L. Cooper, Alexander Joel D. Alon, Monika Akbar, Michael Stewart, Sean Ponce and Stephen H. Edwardsacm Transactions on Computing Education, Vol. 10, No. 3, Article 9, Pub. date: August 2010.

[4] “Visualizing sorting algorithms” by Brian Faria, Rhode Island College, 2017.

[5] “ViSA: Visualization of Sorting Algorithms” by Tihomir Orehovački, ResearchGate, May 2012.

[6] “ Finding the shortest path in a graph and its visualization using C# and WPF” by Radoslav Mavrevski, Metodi Traykov, Ivan Trenchev, International Journal of Computers, ISSN: 2367-8895, Volume 5, 2020.

[7] “Visualization of Abstract Algorithmic Ideas” by Ludˇek Kuˇcera, Proceedings of the 10th International Conference on Computer Supported Education (CSEDU 2018).

[8] “Willow: A Tool for Interactive Programming Visualization to Help in the Data Structures and Algorithms Teaching-Learning Process” by Pedro Moraes and Leopoldo Teixeira, SBES 2019, September 23ś27, 2019, Salvador, Brazil.