



Graph Algo Visualizer

Sweeta Bansal^{1,*}, Karan Kohli¹, K. K. Vishwakarma¹, Kush Gupta¹

¹Computer Science and Engineering, Inderprastha Engineering College, Uttar Pradesh, India
Emails: sweeta.bansal@ipecc.org.in; karan3020000@gmail.com; 21krishna22@gmail.com;
kushgupta.potus@gmail.com

Abstract

Visual aids can be used to visualize different concepts in computer science. Algorithm recognition has been a major theme for Computer Science education for years, but it has not reached schools / teaching halls as a major educational tool. The Graph Algo Visualizer is based on the Design & Analysis of Graph Algorithms. It is a Visualizer that helps to understand and evaluate the working & efficiency of Graph Algorithms easily as compared to the naïve approach of understanding algorithms through code.

Keywords: Graph Visualizer; Data Structure; Data visualization

1. Introduction

The knowledge of Computer Science is incomplete without Data Structures and Algorithms and it forms an essential part of the skillset for any Computer Science professional be it Computer Scientist or a Programmer in a corporate environment and this justifies the subject holding a stable position in the Computer Science curriculum. The ever-growing number of students in higher education requires the introduction of appropriate approaches to their education system. One such useful technique for learning any concept is Visualization. Visualization of Algorithms aids in improving Computer Science Education. The process of teaching and learning algorithms possesses complexity and difficulty on the part of the learner. In this project, we visualize Graph Algorithms.

The project is divided into three modules, namely:

1.1 Path Finding Algorithms: • Dijkstra Algorithm • A* Algorithm (AI-Based Algorithm) • Depth FirstSearch • Breadth-First Search

1.2 Task Scheduling via Topological Sorting

Graph Algo Visualizer aims to help a learner visualize and hence understand the working and performance to be able to apply them in different circumstances as opposed to the conventional method of memorization of code prevalent among students of Computer Science Courses. It also aims to sow the seed of innovative thinking in the minds of young learners who can apply efficient algorithms for the desired application using the imagination power and understanding of algorithms gained with the help of this project. The Graph Algorithms visualized here have many applications not limited to :

- Path Finding: Google Map Routing, Ola/Uber/Swiggy/Zomato Driver Route Prediction, mapping, computer networking, social media.
- Minimum Spanning Tree: Cluster Analysis, Network Design, Protocols in computer science to avoid network cycles, Real-time face tracking, and verification.
- Topological Sorting Finding the presence and resolution of deadlocks, critical path analysis, dependency resolution.

2. Literature Survey

Algorithms & Data Structures are an important but complex part of computer science education for students. One ray of light in the direction of improving learning in this area is to include visualizations and animations in the curriculum. Algorithm Visualization has a rich history in computer science education dating from the 1980s: attempt by Ronald Baekar and the Balsa program [Brown and Sedgewick, 1984]. Since then, hundreds of Exhibits have been introduced and sometimes provides free of charge to teachers. Although many Algorithm Visualizers exist and are freely available on the internet, only few are of high quality and of true value from which a student can actively learn and interact. A Good Algorithm Visualizer can give a new life to the algorithm by the graphical representation of their various states and intermediate transitions independent of their memory interactions. Algorithm Visualizers have been consistently liked by students and educators and some studies have even proved that they can indeed improve understanding of Data Structures & Algorithms. The history of Algorithm Visualizers can be divided into two parts: Those which were implemented before Java and those which were implemented after.

2.1 Prior to JAVA:

- Pioneer of Algorithm Visualizers: Brown Algorithm Simulator and Animator (Balsa) has introduced a program produced by animators to teach essential Computer Science and can be used to eliminate error and research.
- XTANGO[1992] , POLKA[1992] , Swan [1996] are some of the Visualizers of this period.

2.2 After JAVA:

Vast majority of Algorithm Visualizers since the mid-1990s have been implemented using Java with the support of its graphical libraries. For e.g. : JAWAA , JHAVE , ALViE. Various Algorithm Visualizers have been implemented for algorithms and simple data structures such as sorting algorithms like bubble sort, insertion sort, radix sort, etc. Visualizers for Graph Algorithms are currently into the shaping and this project focuses on them. The Visualizer Algorithm can provide another logical and attractive alternative to written teaching methods such as pseudo code and real code. It has been identified that Algorithm Visualizers can be good for:

- Teaching the Concept
- Exploring the concept
- Debugging
- Comparisons
- Lecture Aid
- Lab Exercise
- Self-Study

Although many projects and research papers are dedicated to this field, Algorithm Visualization is still far from the mainstream in learning algorithms and Computer Science Education Tools.

3. Methodology

The proposed methodology for the successful completion of the project is the Iterative Enhancement Model.

The project is divided into three modules and the completion of all three modules marks the completion of the project.

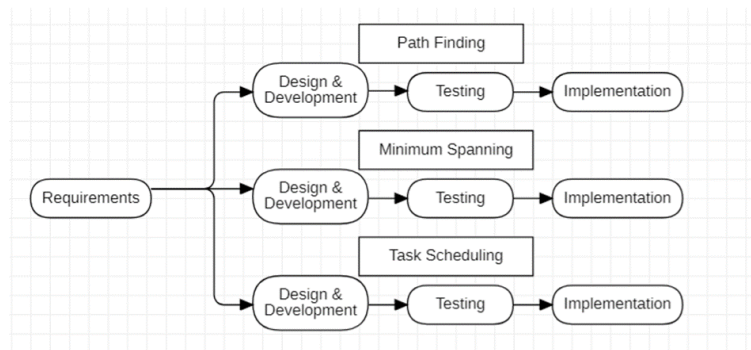


Figure 1: Iterative Enhancement Model

4. Results and Discussions

Modules implemented are as:

4.1 Path Finding Visualizer

As the name suggests this Module focuses on Graph Algorithms that search for a path connecting two points in Network represented as a 2D Grid. Here each movement from one Node to another “costs” 1.

The data

- Path Length
- Path Cost
- Time Complexity

Calculated is based upon the Algorithm Chosen to be visualized and helps to compare the performance of different Algorithms with Each other.

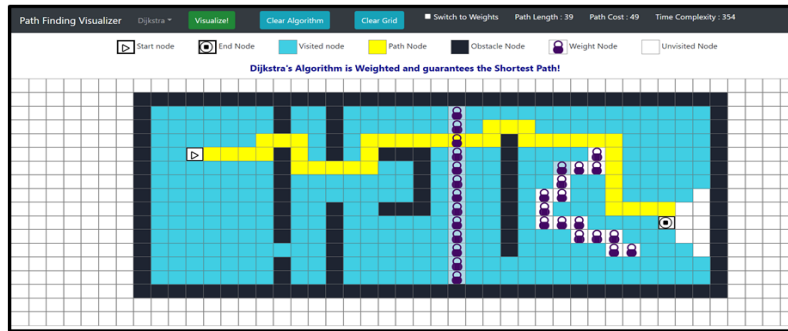


Figure 2: Path Finding Visualizer

5. Minimum Spanning Tree

This Module demonstrates the formation of a minimum spanning tree with the insertion of each node.

Here we used Prim's Algorithm to create a Minimum Spanning Tree. With the insertion of each Node on the Canvas, Prim's Algorithm is applied on all the nodes in the canvas according to which these nodes are connected.

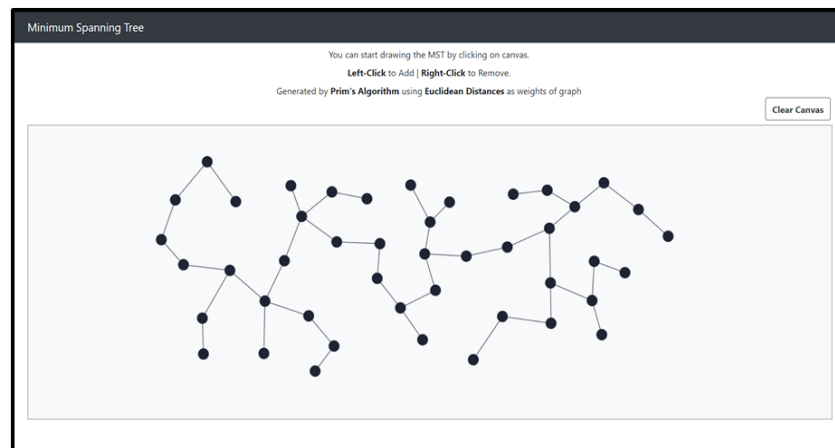


Figure 3: Minimum Spanning Tree

6. Topological Sorting

This module visualizes a direct application of Topological Sorting which is Task Scheduling. To implement Task Scheduling we have used Kahn's Algorithm for Topological Sorting. The user can add an endless number of Tasks and their Prerequisite through entering them in input boxes.

Kahn's Algorithm considers this data as vertices of a graph such that the prerequisite task directs towards the task and creates a Graph according to it in the Canvas.



Figure 4: Topological Sorting

7. Conclusion

With the entirety of this challenge, we have efficaciously performed the objective of our assignment that is to embed some of the Graph Algorithms with Visualization and evaluate their performance or applications. As is the case with maximum different coaching areas, there was a big hole among the principle and sensible expertise of algorithms attention. The primary goal of the project is to use it from research educators and students for teaching and studying the present recognized combinatorial graph algorithms. The idea of the device is to provide an integrated educational environment for both instructors and students to facilitate the learning method in an efficient manner.

References

- [1] Aakansha N. Tabhane, Nikhil Likhar, KalyaniMohod, ManaliKadbe, “Path Finding Visualizer”, IJARIE, Vol-7 Issue-3 2021.[Online serial].
- [2] Nishant Kumar and NidhiSengar, “Pathfinder Visualizer of Shortest Paths Algorithms”, International Journal for Modern Trends in Science and Technology, 6(12): 479-483, 2020.
- [3] Martin Eklund and David Svantesson, “A naive implementation of Topological Sort on GPU”, CSC, KTH, 2016.
- [4] Ahmad AffandiSupli, NorshuhadaShiratuddin and SyamsulBahrinZaibon, “Critical Analysis on Algorithm Visualization Study”, International Journal of Computer Applications (0975 – 8887), Volume 150 – No.11, Sep 2016.
- [5] Navadeepika.K.M.R, Dr.V.D.Ambeth Kumar, “Online Conversion of Tamil Language into Tactile for Visual Impaired People”, International Journal of Advanced Science and Technology Vol. 29, No. 5, pp. 3401 - 3407, (2020).
- [6] Christopher D.Hundhausen, Sarah A.Douglas And John T.Stasko, “ A Meta-Study of Algorithm Visualization Effectiveness”,Journal of Visual Languages &Computing,Volume 13, Issue 3, pp. 259-290, Jun 2002.
- [7] C. Shaffer. A Practical Introduction to Data Structures and Algorithm Analysis.Prentice Hall, second edition, 2001.

- [8] M. Anitha, V. D. Ambeth Kumar, S. Malathi, V. D. Ashok Kumar, M. Ramakrishnan, Abhishek Kumar, and Rashid Ali, “ A survey on the usage of pattern recognition and image analysis methods for the lifestyle improvement on low vision and visually impaired people”, *Pattern Recognition and Image Analysis*” Journal and will be published in Issue 1, Vol. 31, 2021. DOI:10.1134/S105466182101003X
- [9] P. Saraiya. Effective features of algorithm visualizations. Master’s thesis, Department of Computer Science, Virginia Tech, July 2002.