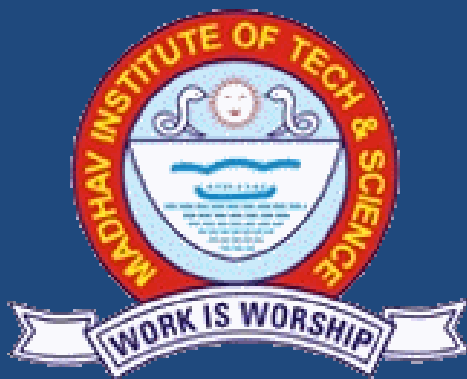


MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR
(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to
RGPV, Bhopal)



DEPARTMENT OF CSE & IT

**Minor Project II
(150606)**

REPORT FILE

Session Jan-July, 2021

SUBMITTED TO:

**Prof. Jamvant S. Kumare
Prof. Amit Manjhar**

SUBMITTED BY:

**ANANT GUPTA (0901CS181007)
JAYANT GANGWANI (0901CS181046)
CSE III YEAR (6th sem)**

Algorithm Visualizer

Pathfinding and Sorting Visualizer

Contents

- **Abstract**
- Purpose of Study
- Methodology Used
- Summary
- **Introduction**
 - Problem Definition
 - Project Overview
 - Hardware Specification
 - Software Specification
- **Literature Survey**
 - Existing System
 - Proposed System
-
- **System Analysis and Design**
 - Requirement Specification
 - Flowcharts/DFDS
 - Design & Test Steps/Criteria
 - Algorithm & Pseudo Code
 - Testing Process
- **Results / Outputs**
- **Conclusions**
- **References**

Abstract

Algorithms are used in every project. We have worked upon some well-known algorithms(asked in our company interviews) and tried to explain them.

Visualizations of algorithms contribute to improving computer science education. The process of teaching and learning of algorithms is often complex and hard to understand problem. Visualization is a useful technique for learning in any computer science course. In this paper an e-learning tool for shortest paths algorithms visualization is described. The developed e-learning tool allows visualizing the algorithm steps execution. It is intended to be used as a supplement to face-to-face instruction or as a stand-alone application. The conceptual applicability of the described e-learning tool is illustrated by implementation of various Algorithms like Dijkstra algorithm,A STAR algorithm,BFS, Greedy Best FIRST search and various Sorting algorithms like INSERTION SORT,Bubble Sort, Merge Sort,QuickSort,Heap Sort etc... The preliminary test results provide evidence of the usability of the e-learning tool and its potential to support students' development of efficient mental models regarding shortest paths algorithms. This e- learning tool is intended to integrate different algorithms for shortest path determination.

We made a system which helps us to understand this algorithms visually i.e we take the sample input ,run the algorithm on that input and with the help of animations, see the algorithm working visually.

We have developed a platform to Visualize pathfinding and Sorting Algorithms using **Reactjs**. This is a web application based project with Animations in it.

By working on this project we came across a lot of new things among which react-animations was one of our major finding and it was so tempting that we used it in our project.

It also Various React Tools Like LAZY and suspense to handle server side rendering before the actual component is mounted over DOM.

And also React with custom Styled Components

Problem Definition:

Algorithms are used in every project. We have worked upon some well-known algorithms(asked in our company interviews) and tried to explain them .Lets take an overview of the project to know how

Algorithm visualizer is a web app written in React.It contains UI Components and Interprets command into visualization.

Server Serves the Web App and provide the API's that I need on the FLY.(e/g – Github SignIN / Compiling Running Code/etc)

Project Overview

Algorithm contains visualizations of algorithms shown in the web App.

Graphical and dynamic web based tools are more appealing for students than traditional learning materials. It has been confirmed that learners spend much more study time when visualization is involved; however, there has been some skepticism about the real value of visualizations as a pedagogical tool. Many educators think that visual tools enhance their lectures and significantly increase student's comprehension, but such tools are of little effectiveness when students are not actively engaged in the learning process

So Basically there are 2 Modules in this Project→

- i. Sorting Visualizer= This project sorting visualizer is a very simple UI and it allows the users to select the sort algorithm, select the array size, and speed of the visualization.

Q- *What's inside the Project Part1*

Currently available sorting algorithms- Bubble sort, Selection sort, Insertion sort, Merge sort, Quicksort, Cockail , Heap, Linear (I will plan to bring more algorithms in action to visualize & more changes).

You can change the size of the array

You can change the speed of the visualization

Writing sort algorithms in JS

And coming to the second challenge I faced writing sort algorithms in JS. When I want to write some code or learn algorithms for solving coding problems, I usually prefer Python/Java. Initially, I took more time to implement the first sort algorithm i.e., Bubble sort to perfectly work. After that, for the remaining algorithms, I took no time and implemented them so quickly one after the other.

High Level Approach→

Creating the website's User Interface (UI) using HTML, CSS and enhancing it further using Bootstrap; without actually implementing any of the app's core features.

Implementation of animations, effects and core functionalities (sorting algorithms) using JavaScript.

Publish ON Github

2>Pathfinder Visualizer=

Pathfinding or pathing is the plotting, by a computer application, of the shortest route between two points. It is a more practical variant on solving mazes. This field of research is based heavily on Dijkstra's algorithm for finding the shortest path on a weighted graph.

Pathfinding is closely related to the shortest path problem, within graph theory, which examines how to identify the path that best meets some criteria (shortest, cheapest, fastest, etc) between two points in a large network. At its core, a pathfinding method searches a graph by starting at one vertex and exploring adjacent nodes until the destination node is reached, generally

with the intent of finding the cheapest route. Although graph searching methods such as a breadth-first search would find a route if given enough time, other methods, which "explore" the graph, would tend to reach the destination sooner. An analogy would be a person walking across a room; rather than examining every possible route in advance, the person would generally walk in the direction of the destination and only deviate from the path to avoid an obstruction, and make deviations as minor as possible.

Hence our pathfinding Visualizer Visualises four Algorithms

1-Dijkstra's Shortest PATH

2-Breadth First Search

3-Greedy Best First Search

4-A Star Algorithm

Hardware Specification

Processor	Pentium 4 or Intel i3 or higher
System	Multimedia PC
RAM	256MB
Hard Disk Space	0.5GB

Software Specification:-

Operating System	Windows
Editor	Visual Studio Code
Browser	Google Chrome
Starter Package	Node js
Pushing the code	Github

LITERATURE SURVEY

Over the years, there have been many studies and papers on the use of sorting algorithms as visual aids. Some are comprehensive views on how to create animations and perform statistical analysis, and others focus on different techniques aimed for increased understanding of a similar animation. By similar, I mean that between two studies, the animation used may be similar, but the difference in analysis was geared to how the

algorithm was used. I would like to say upfront that John T. Stasko was a prominent figure in researching animations and most of my sources include papers in which he contributed. The paper “Algorithm Animation,” by A. Kerren and J. Stasko [3] is a step-by-step guide to analyzing the environment, means, and available coding methods to use a sorting animation. Many different types of software are listed to be used for animation, one of which was BALSAs, which pioneered the interesting event approach [3, p. 3]. BALSAs was created by Marc Brown and the interesting event approach was coined to determine what part of the sorting algorithm was significant to both clearly see and understand how the algorithm performs. Additional software that followed this principle were Zeus, CAT, Tango, and Samba, to name a few [3, p. 2]. These developments prove the continuing interest in creating animation tools. For example, the interesting event approach I took focused on animating the movement of data as a visual description of the algorithm

SYSTEM ANALYSIS & DESIGN

Requirement Specification

The main objective was to help students to understand the well-known and mostly asked algorithms in visual manner or we can say through animations. We require knowledge of a front-end

library react and how to do animations with css. However there is to be something that changes input ,speed, color, etc..

System Architecture :-

The code consists of HTML,CSS, JAVASCRIPT combined in React. React uses JSX which mains HTML and javascript can be written together. There is a large component known as Home which contains two subcomponents PATH-FINDING and SORTING ALGORITHM VISUALIZER.



The sorting component has two components visualizer controller(on left) and actually visualization taking place on second component(on right).Inside controller,we have different features which are selection of algorithms(merge sort, quick sort,bubble sort, cocktail sort, selection sort, insertion sort and

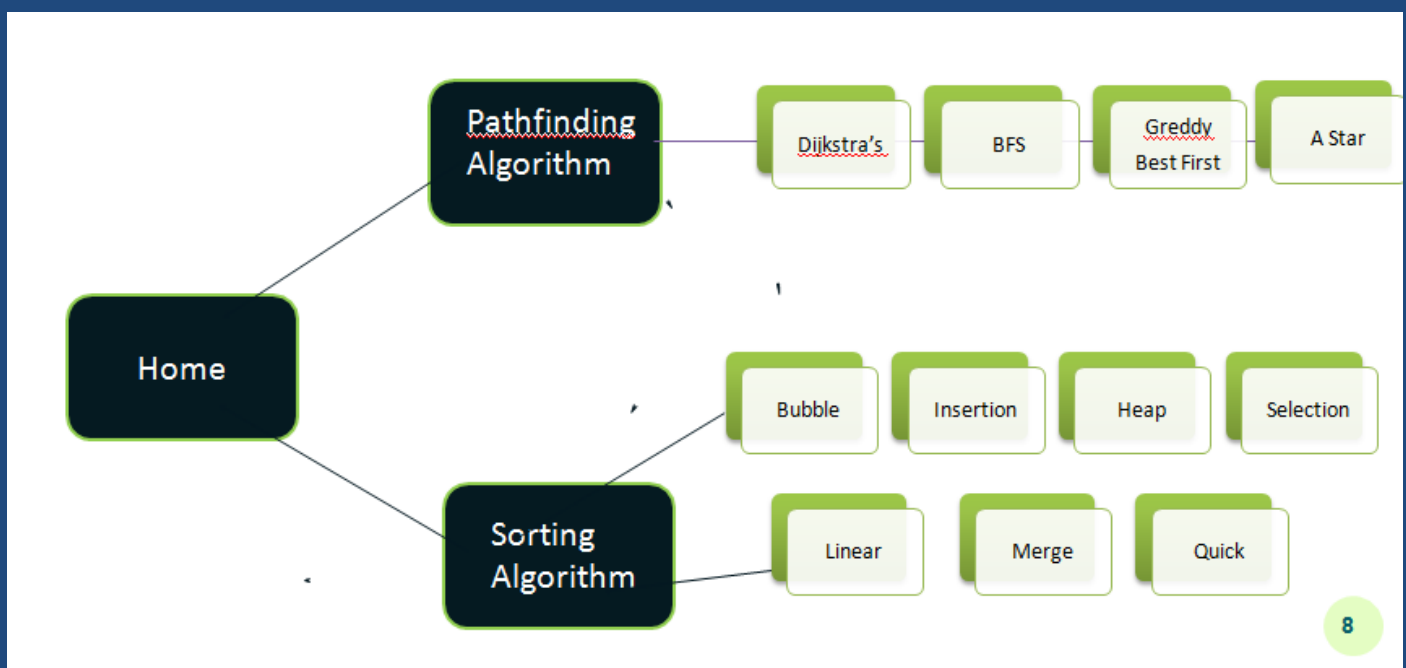
heap sort). Second feature is to select the speed(Very fast, fast, normal, slow, very slow). Third feature is size of array input which we can select from options.Fourth ,fifth and sixth are just colors of bars .

Seventh feature is the main and it just makes a random array of given size. It also stops the execution in between.

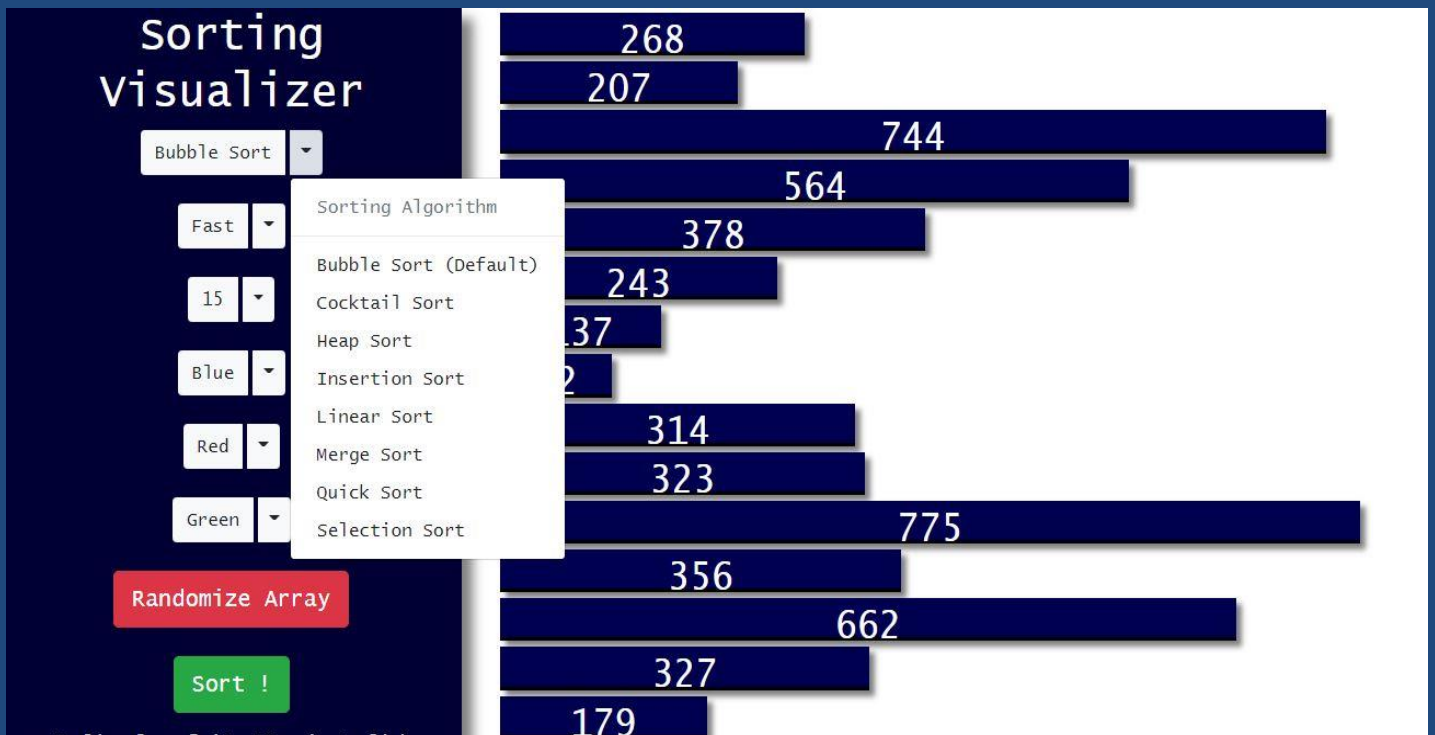
Lastly there is a button which starts sorting.

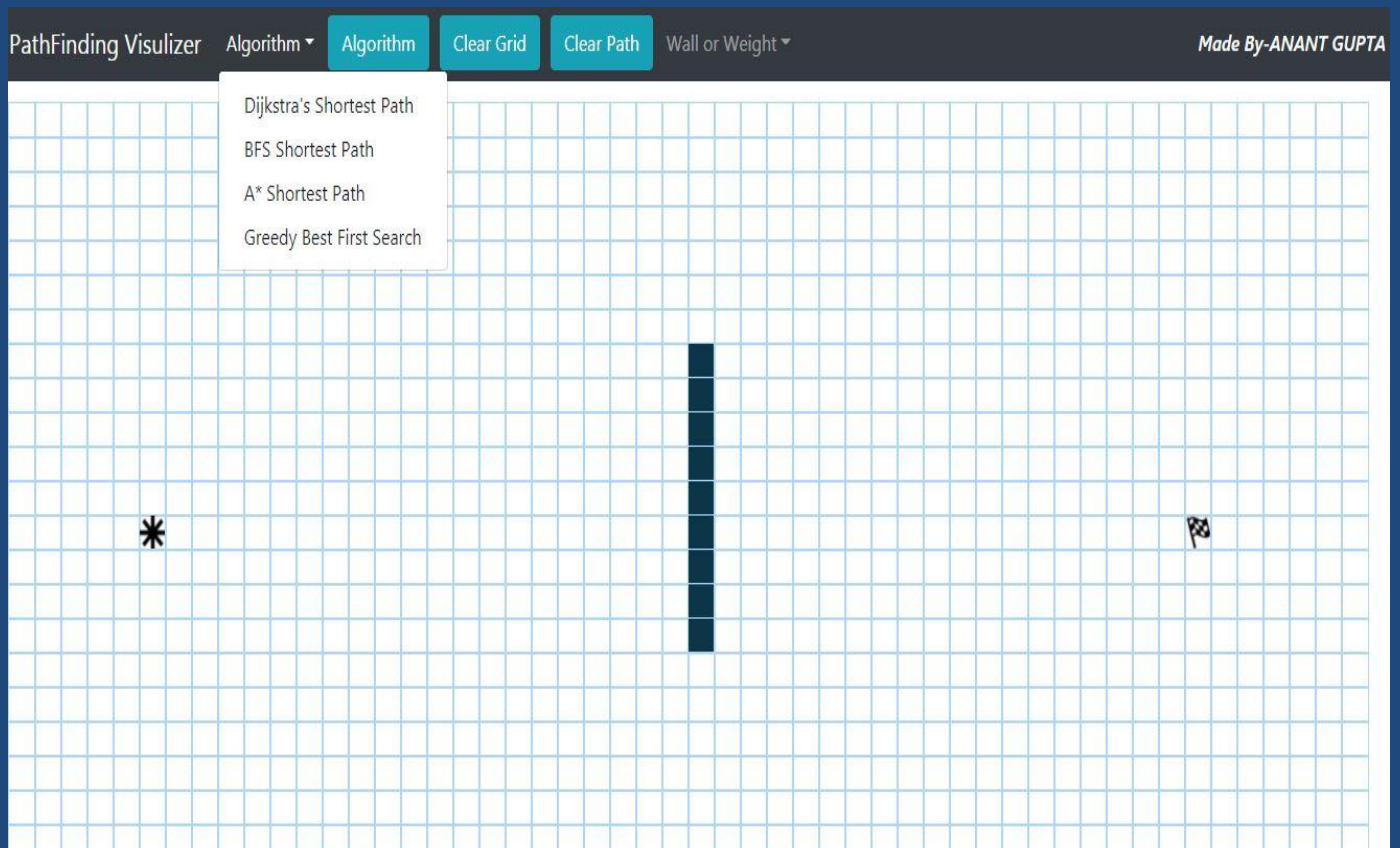
Second main component is Path Finding.It

Flowchart



USER INTERFACE/RESULTS / SCREENSHOTS





CONCLUSION

With the completion of this project, we have successfully achieved our objective of our project is to embed Graph Path Finding with Visualization and Comparing their performance.

As is the case with most other teaching areas, there has been a significant gap between the theory and practical understanding of algorithms realization.

This is true also for shortest paths algorithms and in particular for Dijkstra algorithm,A star ,and BFS.

The main goal of the project is to use it from operations research educators and students for teaching and studying the existing known combinatorial graph algorithms.

The main idea of the system is to provide an integrated educational environment for both instructors and students to facilitate the learning process in efficient way.

To conclude, we have learnt a lot of things working under this project.

REFERENCES

- <https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/>
- <https://medium.com/@urna.hybasis/pathfinding-algorithms-the-four-pillars-1ebad85d4c6b>
- <https://www.geeksforgeeks.org/sorting-algorithms/>
- <https://www.geeksforgeeks.org/bubble-sort/>
- <https://react-bootstrap.github.io/>
- <https://reactrouter.com/web/guides/quick-start>

