**Algorithm Visualizer**

**PROJECT REPORT**

OF MAJOR PROJECT

**BACHELOR OF TECHNOLOGY**

CSE

|  |  |
| --- | --- |
|  |  |
|  | C:\Users\Puneet\Desktop\core-0001-c340d51b1e697990011e77bd1c8154fc.s_group.jpg |

|  |  |
| --- | --- |
| SUBMITTED TO | SUBMITTED TO |

|  |  |
| --- | --- |
| SUBMITTED BY  Vinay Saini (2003294) Vanshika Ahuja(2003287) |  |
|  |  |

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**Chandigarh Engineering College**

**Landran** **Mohali, Punjab - 140307**

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.no** | **Content** | **Page No.** | **Signature** |
| 1. | Introduction | 1-2 |  |
| 2. | Literature Survey | 3-4 |  |
| 3. | Methodology/Planning of work | 5 |  |
| 4. | Facility required for proposed work | 6 |  |
| 5. | References | 6 |  |
| 6. | Project Code | 7-15 |  |
| 7. | Project Output | 16-18 |  |

# Introduction

# In today's world, algorithms play a crucial role in computer software, enabling efficient data organization and retrieval. Sorting algorithms, in particular, are fundamental to the field of computer science, taught to students early on in their studies. As a student myself, I encountered challenges while learning about sorting algorithms during my algorithm design course. It became evident that visual representation is a vital component of the learning process.

# Motivated by the need for a more intuitive understanding of sorting algorithms, I embarked on a project to create a powerful tool for comprehending the inner workings of various well-known sorting algorithms. The ultimate goal of this project was to provide a program that offers an exceptional user experience and serves as a valuable resource for learning and practicing sorting algorithms.

# The program, developed with a user-friendly and intuitive design, allows users to experiment with different sorting algorithms using their own data. Accompanying the program is a comprehensive thesis that describes the principles behind the most widely used sorting algorithms, which are demonstrated within the software. This thesis can be utilized by students as a learning resource, providing a deeper understanding of the algorithms.

# Furthermore, the program is also suitable for use by lecturers and tutors during classes as a visual aid for algorithm demonstrations. It includes programmer documentation and a user guide to facilitate its utilization. Readers of this documentation are expected to have basic programming experience and familiarity with data structures such as arrays, lists, and trees, as well as an understanding of recursive procedures. Additionally, knowledge of simple algorithms and their implementations, along with a grasp of linear algebra and calculus, will enhance comprehension.

# Understanding the differences and similarities between sorting algorithms and knowing when to apply a particular algorithm to a given problem is a skill that requires practice and experience. Throughout my years of studying computer science, I have strived to gain expertise in this area. However, I have found that merely analyzing code does not always provide a complete understanding of how a sorting algorithm functions. While examining time and space complexities helps to grasp the underlying processes, visualizing the decision-making involved in the sorting process is truly transformative.

# This motivation drove me to develop a visualization tool as my final project. I aimed to create a tool that would enable anyone to enhance their understanding of essential sorting algorithms. By implementing and visualizing these algorithms myself, I anticipated uncovering new insights about their inner workings.

# Algorithm visualization is a graphical approach that simplifies and deepens the understanding of how algorithms operate. As algorithms and data structures are integral to computer science education, it is crucial to incorporate appropriate teaching methods. With the increasing number of students pursuing computer science in higher education systems, the integration of algorithm visualizations becomes paramount. In this project, we discuss the extension of traditional teaching methods, such as whiteboards and slides, with algorithm visualizations, particularly within the context of higher education in computer science.

# By combining theoretical knowledge, practical implementation, and visualizations, this project aims to provide a comprehensive and accessible resource for individuals seeking to enhance their understanding of sorting algorithms. Whether you are a student, educator, or programmer, this project offers a valuable tool for exploring and mastering the intricacies of sorting algorithms.

# Literature Survey

Pathfinding Visualizer Project

The Pathfinding Visualizer project, developed by Clement Mihailescu, aims to provide a visual representation of pathfinding algorithms in action. In order to understand the context and related work in this field, a literature survey was conducted, which revealed several relevant studies and papers.

1. "Algorithm Animation" by A. Kerren and J. Stasko:

This paper serves as a comprehensive guide to analyzing the environment, means, and coding methods for creating sorting animations. It discusses different software tools used for animation, such as BALSA, Zeus, CAT, Tango, and Samba. The interesting event approach, pioneered by BALSA, is highlighted as a significant contribution to understanding sorting algorithms through visual representation [3].

2. "Using Student-Built Algorithm Animations as Learning Aids" by J. Stasko:

In this article, Stasko explores the idea of students building their own algorithm animations as a means of enhancing understanding. The visual programming tool Samba is introduced, and students are assigned to create animations that can be easily comprehended by individuals unfamiliar with the material. The results demonstrate improved understanding and the identification of misconceptions when implementing algorithms in animations [5].

3. "Do Algorithm Animations Assist Learning? An Empirical Study and Analysis":

This paper provides an in-depth examination of how students respond to sorting animations. A post-test study is conducted to compare the performance of students who had access to textbook resources only versus those who had both textbooks and animations as learning aids. Although the study does not find clear evidence that animations significantly enhance learning, the group with animations performed slightly better on the post-test [6]

The literature survey reveals that algorithm animations have been a topic of substantial research in education and learning aids. While some studies show only marginal improvements in learning outcomes with animations, recent attempts, such as the Pathfinding Visualizer project, aim to present algorithmic concepts in new ways, such as through videos, to aid individuals struggling with algorithms.

It is worth noting that the literature survey primarily focuses on sorting algorithms and general algorithm animations. However, the Pathfinding Visualizer project specifically deals with pathfinding algorithms. To extend the literature survey for the project, it would be beneficial to explore studies specifically related to pathfinding algorithms and their visualization.

# 

# Methodology/Planning of work

1. Project Overview:

- Provide a brief overview of the Pathfinding Visualizer project.

- Explain its purpose and goals, such as visualizing different pathfinding algorithms.

- Mention the key features and functionalities of the application.

2. Project Setup:

- Explain the necessary steps to set up the development environment.

- Specify the required software and dependencies, such as a code editor and a web browser.

- Provide instructions for cloning or downloading the project repository from GitHub.

3. Front-End Development:

- Describe the front-end development process, which focuses on the user interface.

- Identify the main HTML file (index.html) and its structure, including the <head> tag.

- Explain the role of the "src/" folder, which contains the JavaScript code for visualization logic.

- Detail the visualization techniques used, such as representing elements with pixels or bars.

4. Back-End Development:

- Discuss the back-end development aspects related to the business logic of the application.

- Outline the different pathfinding algorithms implemented in the project.

- Explain how the algorithms work and their relevance to the visualization process.

- Describe any data structures or algorithms used to support the pathfinding algorithms.

5. Integration:

- Discuss the integration of the front-end and back-end components.

- Explain how the visualization logic interacts with the pathfinding algorithms.

- Describe any APIs or libraries used for this integration.

6. Testing and Debugging:

- Explain the testing approach and methodologies used to ensure the application's functionality.

- Detail the types of tests performed, such as unit tests or user acceptance tests.

- Describe any debugging techniques or tools utilized during the development process.

7. Documentation and Deployment:

- Discuss the documentation process, including code comments and README files.

- Provide instructions for deploying the application on a web server.

- Explain any additional considerations for deployment, such as performance optimization or security.

8. Future Enhancements:

- Suggest possible improvements or additional features for the project.

- Discuss potential areas of expansion or optimization.

- Encourage contributors or users to provide feedback and contribute to the project's growth.

9. Conclusion:

- Summarize the overall project implementation methodology.

- Highlight the achievements and outcomes of the project.

- Express gratitude to the project contributors and acknowledge their contributions.

Note: The provided methodology is a general outline. Please adapt it to fit the specific details and requirements of the Pathfinding Visualizer project.

**Facilities required for proposed work**

System requirements:

* Operating system: Linux-Ubuntu 16.04 to 17.10 or windows 7 to 10, with 2GB

RAM(4GB preferable)

* Any browser window with its supported versions

# References

* Geeks for geeks
* W3schools
* Javatpoint

**Project Code**

# HTML CODE

<!DOCTYPE html>

<html>

<head>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384gH2yIJqKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV8i59U5AR6csBvApHHNl/vI1Bx" crossorigin="anonymous">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/js/bootstrap.bundle.min.js" integrity="sha384-

A3rJD856KowSb7dwlZdYEkO39Gagi7vIsF0jrRAoQmDKKtQBHUuLZ9AsSv4jD4Xa" crossorigin="anonymous"></script>

<title>Algorithm visualizer</title>

<link rel = "icon" href = "static/icon.jpeg" type = "image/x-icon">

<link rel="stylesheet" href="css/style.css">

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<div class="container-fluid">

<a class="navbar-brand" href="#">

<img src="static/icon.jpeg" height="25px" class="d-inline-block align-text-top">

Algorithm Visualizer</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span></button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav me-auto mb-2 mb-lg-0">

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" ariaexpanded="false">

Sorting </a>

<ul class="dropdown-menu">

<li><a class="dropdown-item" href="sorting/sorting.html">Bubble Sort</a></li>

<li><a class="dropdown-item" href="sorting/sorting.html">Insertion Sort</a></li>

<li><a class="dropdown-item" href="sorting/sorting.html">Selection Sort</a></li>

<li><a class="dropdown-item" href="sorting/sorting.html">Quick Sort</a></li>

<li><a class="dropdown-item" href="sorting/sorting.html">Merge Sort</a></li>

</ul> </li>

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" ariaexpanded="false">

Searching</a>

<ul class="dropdown-menu">

<li><a class="dropdown-item" href="searching/searching.html">Linear Serach</a></li>

<li><a class="dropdown-item" href="searching/searching.html">Binary Serach</a></li> </ul> </li> <li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" ariaexpanded="false">

Path Finding</a>

<ul class="dropdown-menu">

<li><a class="dropdown-item" href="path finding/index.html">Dijkstra</a></li>

<li><a class="dropdown-item" href="path finding/index.html">A\*</a></li>

<li><a class="dropdown-item" href="path finding/index.html">Gready</a></li>

<li><a class="dropdown-item" href="path finding/index.html">BFS</a></li>

<li><a class="dropdown-item" href="path finding/index.html">DFS</a></li> </ul></li>

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" ariaexpanded="false">

Backtracking </a>

<ul class="dropdown-menu">

<li><a class="dropdown-item" href="backtracking/suduko.html">Sudoku</a></li>

<li><a class="dropdown-item" href="backtracking/backtracking.html">N-Queens</a></li></ul> </ul> <div class="dropdown">

<button class="btn btn-success dropdown-toggle ml-50" type="button" data-bs-toggle="dropdown" ariaexpanded="false">

Quiz Time </button>

<ul class="dropdown-menu ml-50">

<li><a class="dropdown-item" href="sorting/quiz.html">Sorting</a></li>

<li><a class="dropdown-item" href="searching/quiz.html">Searching</a></li>

<li><a class="dropdown-item" href="path finding/quiz.html">Path Finding</a></li>

<li><a class="dropdown-item" href="backtracking/quiz.html">Backtracking</a></li></ul></div></div></div>-</nav>

<header class="overflow-hidden" style="padding-top: 7%;">

<h1 class="text-center text-monospace text-capitalize ">Hello Learners</h1>

<h4 class="text-center">Visualize Algorithm for a better understanding</h4>

<p class="text-center">Click one of the categories below to visualize algorithm.</p> </header>

<div class="row row-cols-2 row-cols-md-3 g-4 justify-content-center">

<div class="col">

<a href="searching/searching.html"> <div class="card text-center" style="width: 18rem;">

<img src="static/search.jpg" class="card-img-top" alt="..." height="160px">

<div class="card-body">

<p class="card-title">Searching</p></div><div></a></div>

<div class="col">

<a href="sorting/sorting.html"><div class="card text-center" style="width: 18rem;">

<img src="static/soting.png" class="card-img-top" alt="..." height="160px">

<div class="card-body">

<h5 class="card-title">Sorting</h5> </div>

</div></a></div>

<div class="col">

<a href="path finding/index.html"><div class="card text-center" style="width: 18rem;">

<img src="static/path\_finding.jpeg" class="card-img-top" alt="...">

<div class="card-body">

<h5 class="card-title">Path Finding</h5></div> </div></a></div>

<div class="col">

<a href="backtracking/suduko.html"><div class="card text-center" style="width: 18rem;">

<img src="static/backtracking.png" class="card-img-top" alt="...">

<div class="card-body">

<h5 class="card-title">Backtracking</h5></div> </div></a></div></div>

<footer class="footer">

<p>Copyright @2022<br></footer>

<script src="script/sorting/main.js"></script>

<script src="script/sorting/script.js"></script>

<script src="script/sorting/sort.js"></script>

</body>

</html>

# CSS Code

.dropdown-item button{ border: none; background-color: white;} #custamization{ display: flex; margin-top: 0.5%; background-color: #eeeeee;}

#array-inputs div{ width: 75%;} #array-inputs { width: 75%;} #array-range{ display: flex;}

#array-inputs button{ display: block; position: relative; left: 15%; margin-top: 25px; margin-bottom: 1%; cursor: pointer;

transition: all 0.3s ease-in-out; background-color: #e9eded;} #options{ display: inline-flex;} #element{

align-self: center; margin-left: 40%; margin-top: 1.5%;} #search\_element{ display: flex;} .dropbtn{ padding: 16px; font-size: 16px; border: none; cursor: pointer; border-radius: 15%;} .algo-dropdown{ margin-top: 0.5%; position: relative; display: inline-block;} .dropdown-content{ display: none; background-color: white; position: absolute; min-width: 130px; box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2); z-index: 1;}

.dropdown-content button{ background-color: white; color: black; padding: 12px 16px; text-decoration: none; display: block; border: 0ch;} .show{

display: block;} #array\_container{ background-color:#D7D7D7;

display: flex; margin-inline: 13vw; width: 75%; height: 65vh;} .align{ vertical-align: bottom;} #popup\_container{ margin-top: 2%; display: flex; justify-content: center;} #btn{ padding: 10px 20px; background: #fff; /\* border: 0; \*/ outline: none; cursor: pointer; font-size: 22px; font-weight: 500; border-radius: 30px;} .popup{ width: 500px; border-radius: 6px; position: absolute; top:0; left: 50%; transform: translate(-50%,-50%) scale(0.1); text-align: center; padding: 0 30px 30px; visibility: hidden; transition: transform 0.4s, top 0.4s;}

.open\_popup{

visibility: visible; top:50%; transform: translate(-50%,-50%) scale(1);}

.header{ display: inline-flex; align-items: center;} #time\_complexity{ font-weight: bold;} #space\_complexity{ font-weight: bold;} #close\_button{ top: 12%; position: absolute; left:90%; border: none;

background: transparent;} #popup h2{

font-size: 38px; font-weight: 500; margin: 30px 0 10px;} #ok\_button{

cursor: pointer; width: 75%; border-radius: 6px; margin-top: 50px; padding: 10px 0; outline: none; font-size: 18px; box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);}

.active{ filter: blur(10px); pointer-events: none; user-select: none;} header{ background-image: url(../static/background.jpeg); background-size: cover; background-position: center; background-attachment: fixed; height: 350px;} .nav-item{ margin-right: 30px;} .card{ margin-top: 2rem; margin-left: 2.5rem;} .footer{ font-size: x-large; margin-top: 15px; background-color: black; text-align: center; color: white;} .ml-50{ margin-left: -50%;}

# Script Code

function myFunction(){ document.getElementById("myDropdown").classList.toggle("show");} window.onclick=function(event){ if(!event.target.matches('.dropbtn')){ var dropdowns=document.getElementsByClassName("dropdown-content"); var i;

for (i=0;i<dropdowns.length;i++){ var openDropdown=dropdowns[i]; if(openDropdown.classList.contains('show')){ openDropdown.classList.remove('show'); }} }} var inp\_size=document.getElementById('array\_size'),arraySize=inp\_size.value; var inp\_gen=document.getElementById('array\_generate'); var inp\_speed=document.getElementById('array\_speed'); var algo\_buttons=document.querySelectorAll(".dropdown-content button"); var algo\_dropdown=document.querySelectorAll(".dropdown-item button") var div\_sizes=[]; var divs=[]; var margin\_size; var cont=document.getElementById("array\_container"); cont.style="flex-direction:row; vertical-align:bottom;"; inp\_gen.addEventListener("click",generate\_array); function generate\_array(){ cont.innerHTML=""; for(var i=0;i<arraySize; i++){ div\_sizes[i]=Math.floor(Math.random()\*0.5\*(inp\_size.max-inp\_size.min))+10; divs[i]=document.createElement("div"); cont.appendChild(divs[i]); margin\_size=0.1; divs[i].classList.add("align"); divs[i].innerHTML=div\_sizes[i]; divs[i].style=" margin:0% " + margin\_size + "%; background-color:#1C6F75; width:"+(100/arraySize-(2\*margin\_size)) +

"%; height:" + (div\_sizes[i])+ "%; vertical-align:bottom"; }} inp\_size.addEventListener("input",update\_array\_size); function update\_array\_size(){ arraySize=inp\_size.value; generate\_array();} window.onload=update\_array\_size(); var algo\_name;

for (var i=0;i<algo\_buttons.length;i++){ algo\_buttons[i].addEventListener("click",algoCall);} for(var i=0;i<algo\_dropdown.length;i++){ algo\_dropdown[i].addEventListener("click",algoCall);} function disable\_buttons(){ for(var i=0;i<algo\_buttons.length;i++){ algo\_buttons[i].disabled=true; inp\_size.disabled=true; inp\_gen.disabled=true; }} function algoCall(){ disable\_buttons(); switch(this.innerHTML){ case "Bubble Sort":Bubble();

break;

case "Insertion Sort":Insertion();

break;

case "Selection Sort":Selection\_sort();

break;

case "Quick Sort":Quick();

break;

case "Merge Sort":Merge();

break; }}

var blur=document.getElementById("array\_container") var popup=document.getElementById("popup"); function open Popup(){ popup.classList.add("open\_popup"); blur.classList.add('active');} function closePopup(){ popup.classList.remove("open\_popup"); blur.classList.remove('active');}

var speed=130;

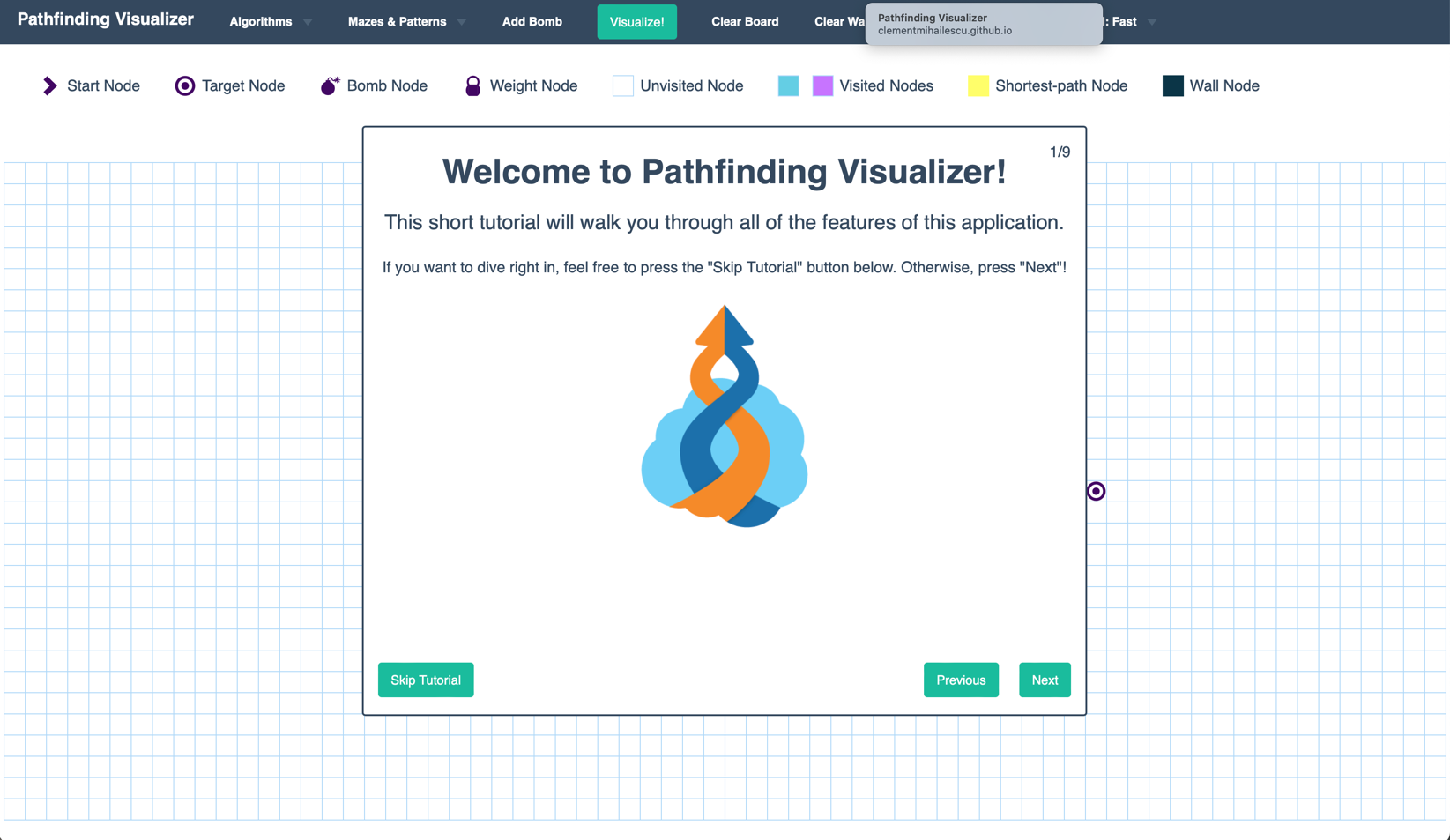
inp\_speed.addEventListener("input",manage\_speed); function manage\_speed(){ var a\_speed=inp\_speed.value; switch(parseInt(a\_speed)){ case 1: speed=70; break; case 2: speed=100; break; case 3: speed=130; break; case 4: speed=150; break;

case 5: speed=170; break; }delay\_time=10000/(Math.floor(arraySize/10)\*speed);} var delay\_time=10000/(Math.floor(arraySize/10)\*speed); var c\_delay=0; function vis\_div(cont,height,color){ window.setTimeout(function(){ cont.style="margin:0% "+ margin\_size + "%; width:" + (100/arraySize-(2\*margin\_size)) + "%; height:" + height +"%; background-color:" + color + ";"; cont.innerHTML=height;},c\_delay+=delay\_time)} function enable\_buttons(){ window.setTimeout(function(){ for(var i=0;i<algo\_buttons.length;i++){ algo\_buttons[i].disabled=false; inp\_size.disabled=false;

inp\_gen.disabled=false; } },c\_delay+=delay\_time);}

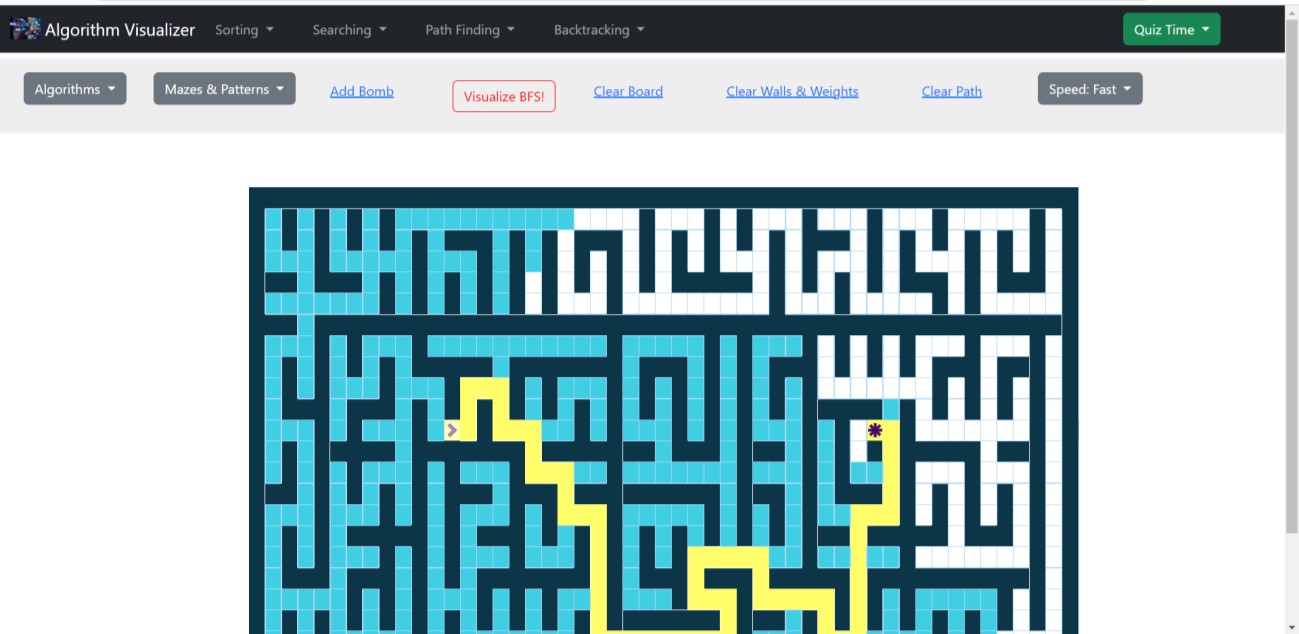
**Output:**

1. FrontPage

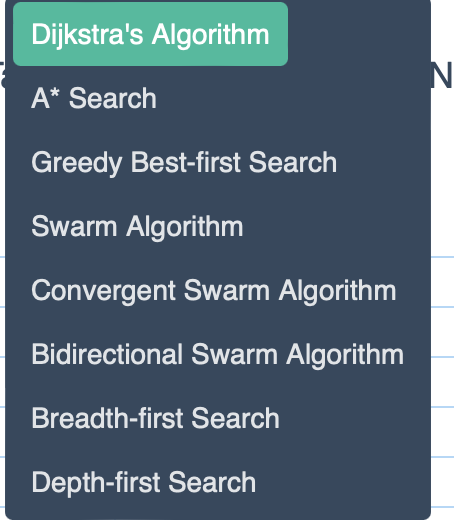


2.) SORTING ALGO

1. Path Finding



3.) Algorithms



4.) MAZE AND PATTERN

A picture containing text, screenshot, font

Description automatically generated