Algorithm Visualizer

PROJECT REPORT

OF MAJOR PROJECT

BACHELOR OF TECHNOLOGY
CSE

SUBMITTED BY

Vinay Saini (2003294)
Yash Vashisht(2003303)

SUBMITTED TO

Ms. Gagandeep

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

Nowadays sorting algorithms are widely used in computer software. For example, if you open file explorer on your PC, you may see files sorted in different ways. Searching in sorted data is more efficient than in not sorted ones. Students of computer science start learning different algorithms in the first year of studies and sorting algorithms are among them.

Since I faced the problems of sorting during the course of algorithm design in the first year of my studies, there is an understanding that the visual representation is a vital part of the studying process. During working on the it was very exciting to learn different techniques of sorting algorithms into the depth. The main goal of the thesis was to create a program which would serve as a tool for understanding how most known sorting algorithms work. There was an attempt to make the best possible user experience. The demonstration software is made in a user-friendly and easy-to-use style. To gain maximal benefit from learning you can try each sorting algorithm on your data. The text of the thesis describes principles of the most known sorting algorithms which are demonstrated in the computer program. It might be used as a source for learning algorithms by students.

Also, the program might be easily used as a demonstration by lecturers and tutors during classes. Besides, there is programmer documentation and user guide to the provided software. Readers of this text are expected to have some programming experience to know basic data structures such as arrays, lists, trees and understand recursive procedures. Also, knowledge of some simple algorithms and their implementations could be helpful. In order to understand the topic better, knowledge of linear algebra and calculus is involved.

The ability to comprehend the differences and similarities between different sorting algorithms and know when to apply a certain one to a given problem is something

that takes time and practice to possess. In my several years studying topics in computer science I've tried my best to gain this expertise, but from my experience it can be difficult to actually understand what a certain sorting algorithm is doing just by analyzing the code that comes with it. Checking out the time and space complexity of these algorithms can help you to grasp what is actually going on, but actually being able to visualize the decisions being made by the algorithm during the sorting process is a game changer. This is where the motivation came from when I was deciding on what I wanted to develop for my final project. My overall goal for this project was to develop a visualization tool that anyone could use to help them further their understanding of some of the most important sorting algorithms. Also, in the process I figured I would also learn some things about these algorithms that I didn't know before by being the one who actually implements them and creates the visualization.

Algorithm visualization illustrates how algorithms work in a graphical way. It mainly aims to simplify and deepen the understanding of algorithms operation. Algorithms and data structures as an essential part of knowledge in a framework of computer science1have their stable position in computer science curricula2, since every computer scientist and every professional programmer should have the basic knowledge from the area .With the increasing number of students in Central European's higher education systems in last decades more concrete numbers and impacts for the case of Slovak one can be found in introduction of appropriate methods into the process of their education is also required. Our scope here is the higher education in the field of computer science. So within the paper, we discuss the extension of standard methods of teaching algorithms, using the whiteboard or slides, with the algorithm visualizations.

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 BALSA, which pioneered the interesting event approach [3, p. 3]. BALSA 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.

Another article I found is Stasko's "Using Student-Built Algorithm Animations as Learning Aids." [5] It initially stated the same fault found in the previous article, where showing an animation does not promote understanding as much as desired. In an interesting twist, however, students were given assignments to build the animations themselves, rather than use some already made as a means of better understanding. The students were introduced to the visual programming tool Samba.

After a few introductory assignments to get the knack for using the software, they were given the option to animate an algorithm, keeping in mind that a person who did not know the material could understand it. The results showed positive feedback and overall better understanding of how the algorithms worked. Also, some students found that they had misconceptions about the material when implementing them in an animation.

For a direct analysis of how students respond to sorting animation, the paper "Do Algorithm Animations Assist Learning? An Empirical Study and Analysis" [6] provides an in-depth view. A post-test study was used to gather information on comparing the results of students who only had textbook resources to those who had a textbook as well as an animation for assistance. The post-test was the same for each group of students, which covered a comprehensive view of the topic. The study found no clear support that an animation would help students with the material significantly. The group of students that had the animation tool in addition to the textbook material averaged correctly answering two more questions than the control group [6]. The paper concluded that visualizing algorithms sounds good, but may not achieve the desired results when implemented.

Overall, there has been substantial research in different forms to support the idea of algorithm animations for education/learning aids. Even though some of the results showed minimal improvement with the introduction of an animation, there have still been recent attempts to spin the material in new ways that may be helpful to someone struggling with algorithms (like videos).

Methodology/Planning of work

This implementation is broadly divided into 2 steps which is decided based on various factors, considering the facts that all these three steps can be implemented parallel without disturbing the progress of other. Making the development process smooth and fast reaching our end goal is also the important factor which can't be ignored. So belowarethe2major division of this implementation. A. Setting Up The Environment This step is chosen as the first step because is focuses mainly on the front-end rather than the business login.

main index.html file within the <head> tag. src/ is the main folder which consist of visualization logic written in java script

The success of Sorting Out Sorting made sorting algorithms a perennial favorite for algorithm animation. Indeed, the sorting problem lends itself quite naturally to visual presentation via vertical or horizontal bars or sticks of different heights or lengths, which need to be rearranged according to their sizes. This presentation is convenient, however, only for illustrating actions of a typical sorting algorithm on small inputs. Hence for dataset the screen is divided in each pixels which is taken as an element in an array and then the array is shuffled using shuffle(array)

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="sha384-gH2yIJqKdNHPEq0n4Mqa/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" aria-expanded="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" aria-expanded="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" aria-
expanded="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" aria-
expanded="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" aria-
expanded="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);}
```

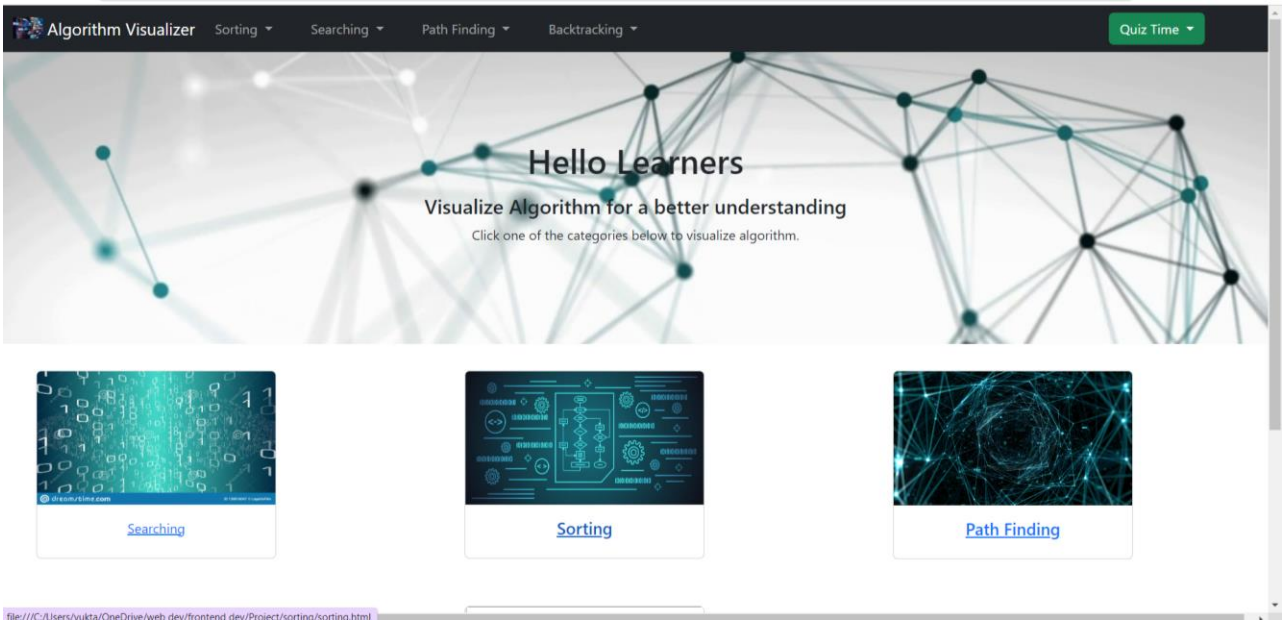
```
for(var i=0;i<algo_dropdown.length;i++){
   algo_dropdown[i].addEventListener("click",algoCall);}
function disable_buttons(){
   for(var i=0;i<algo_buttons.length;i++){
      algo_buttons[i].disabled=true;
      inp_size.disabled=true;
      inp_gen.disabled=true;   }}
function algoCall(){
   disable_buttons();
   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;
```

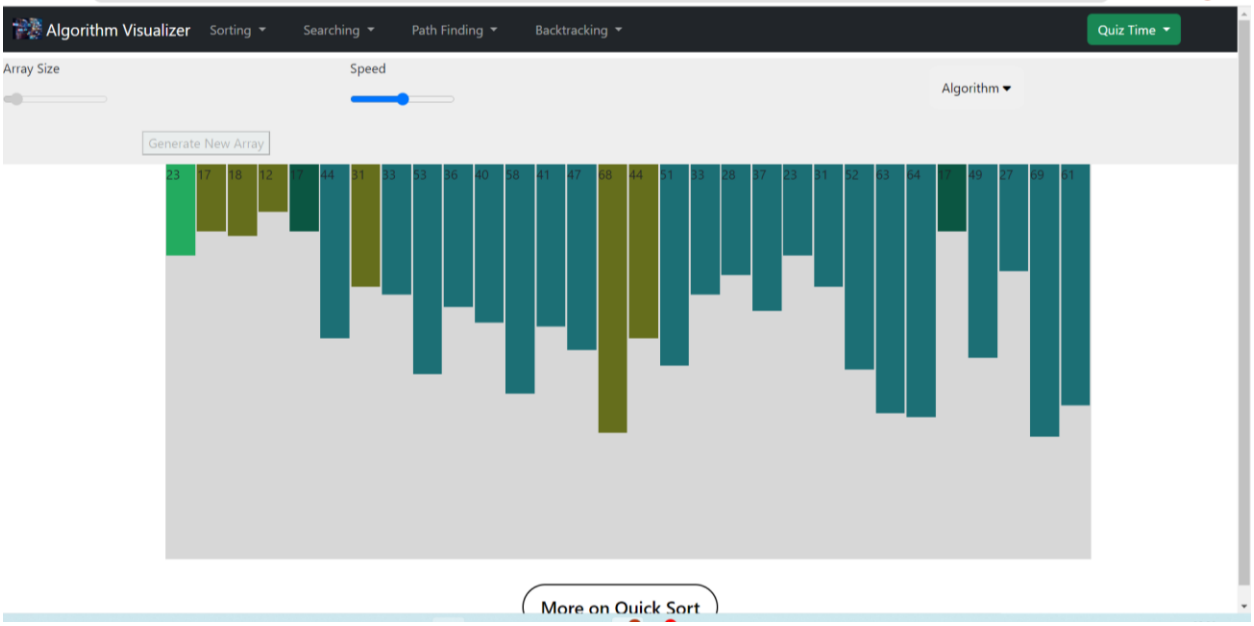
```
        case 5: speed=170;
                break;  }delay_time=10000/(Math.floor(arraySize/10)*speed);}
var delay_time=10000/(Math.floor(arraySize/10)*speed);
var c_delay=0;
function vis_div(cont,height,color){
    window.setTimeout(function(){
        cont.style="margin:0% "+ margin_size + "%; width:" + (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:

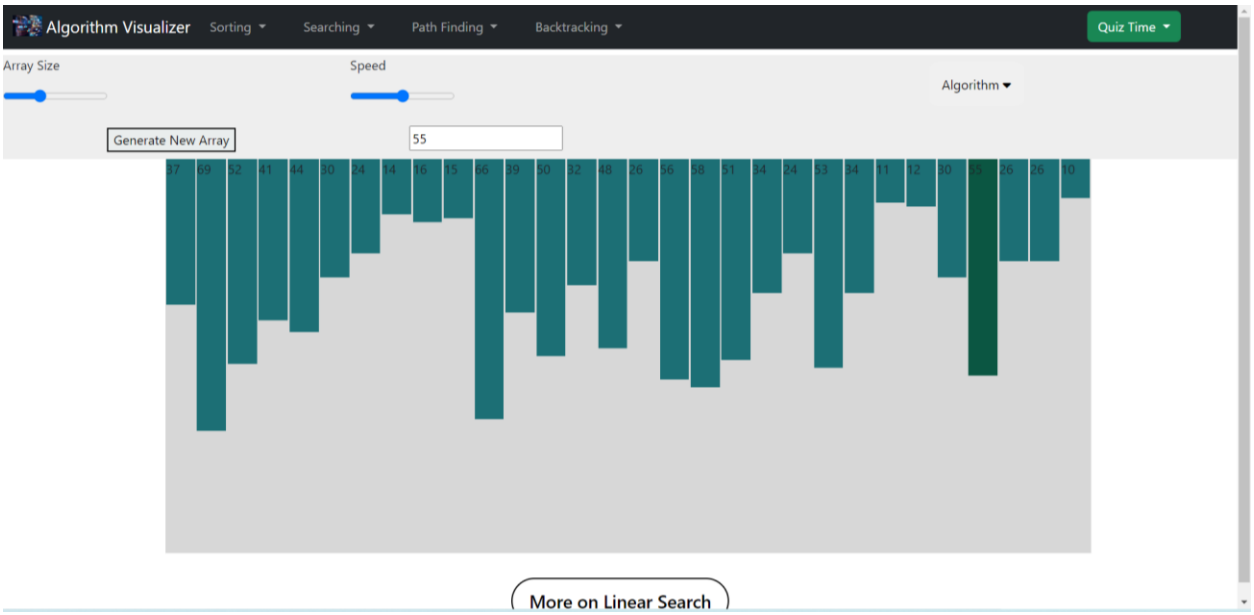
1) Front Page



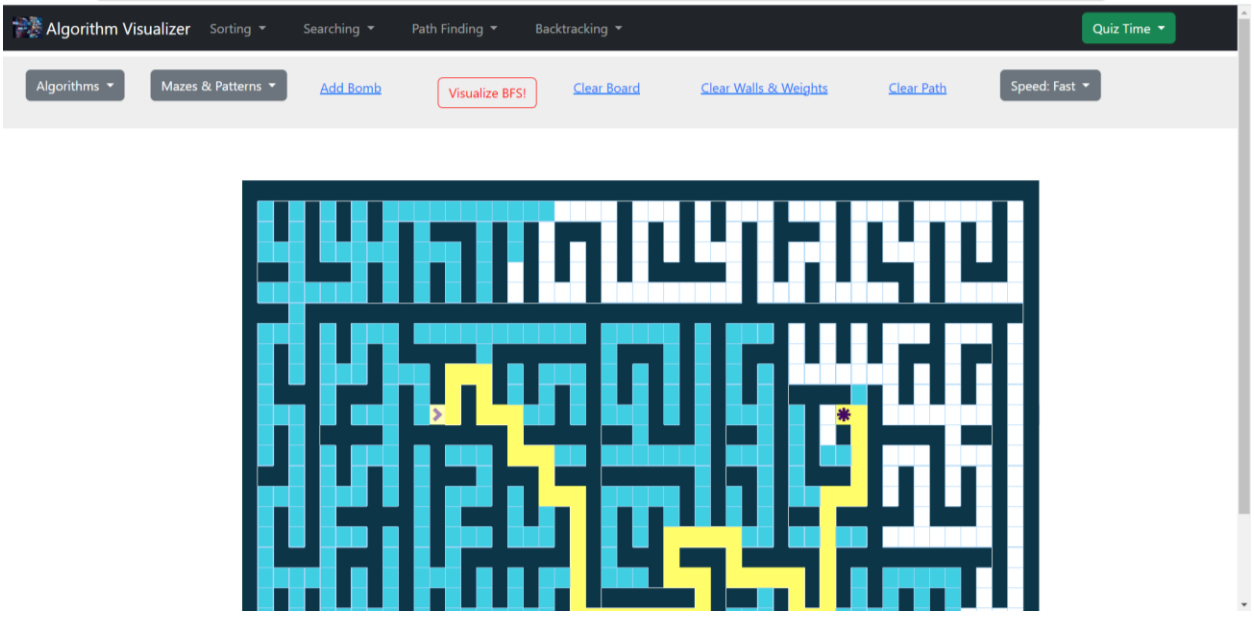
2) Sorting



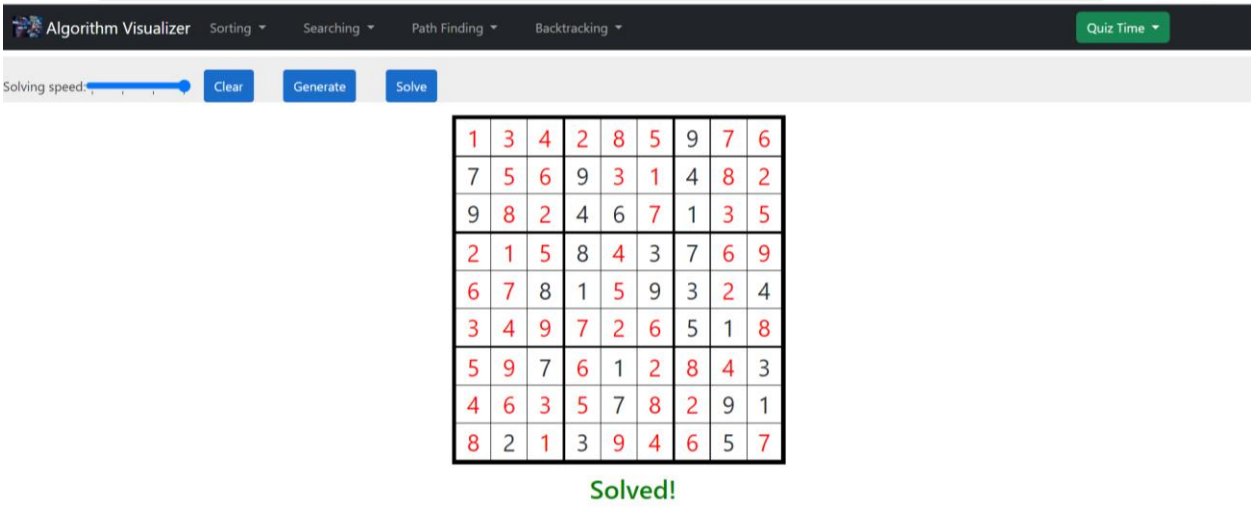
3) Searching



4) Path Finding



5) Backtracking



6) Quiz Time