**Ram Lal Anand College**

**University of Delhi**

**Department of Computer Science**

**Session : 2023 – 2024**

**Project File**

**ISORT- The Sorting Visualizer**

**Course Name – B.Sc. (Hons) Computer Science**

**Semester – V**

**Title of the Paper - Internet Technologies**

**Paper Code - 32341501**

**Name of the Student – Rishabh Chopra**

**Examination Roll No – 21058570041**

**Code**

**HTML FILES -**

**MAIN INDEX FILE-**

<!DOCTYPE html>

<html

<head>

<title>iSort - The Sort Visualizer </title>

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<meta name="language" content="English">

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

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

<link rel= "stylesheet" type= "text/css" href= "css/style1.css">

<link rel= "stylesheet" type= "text/css" href= "css/style2.css">

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

<link rel="icon" href="favicon.ico" type="image/x-icon">

</head>

<body>

<script src="https://unpkg.com/aos@2.3.1/dist/aos.js"></script>

<script type="text/javascript" src="javascript/index.js"></script>

<!-- Side Navigation -->

<div class="sidenav" id="sidenav">

<a href="index.html" class="sidenav-title no-remove"><i class="material-icons icon">home</i> Home</a>

<div id="sep" class="no-remove"></div>

<div class="sidenav-title no-remove"><i class="material-icons icon">sort</i> Sorts</div>

<a href="sorts/bubbleSort.html" class="sidenav-element no-remove">Bubble Sort</a>

<a href="sorts/selectionSort.html" class="sidenav-element no-remove">Selection Sort</a>

<a href="sorts/insertionSort.html" class="sidenav-element no-remove">Insertion Sort</a>

<a href="sorts/countingSort.html" class="sidenav-element no-remove">Counting Sort</a>

</div>

<div class="topnav">

<button class="sidenav-btn topnav-element ripple open" id="sidenav-menu"><i class="material-icons icon">menu</i></button>

<a href="/" class="topnav-element">SORTING VISUALIZER</a>

</div>

<!-- Main text -->

<div id="header"><h1 class="header\_\_title"><center><center>iSORT </center> The Witty Visualizer</center></h1></div>

<div id="cover">

<div id="description-box">

<h2 style="font-family: 'Electrolize', Courier, monospace;">

Sorting Algorithms

</h2>

<div class="description-content">

<p>

In computer science, a <a href="https://brilliant.org/wiki/sorting-algorithms/">Sorting algorithm</a> is an algorithm

that puts a random set of numbers into an ascending or decending sequence.

A Sorting algorithm that puts the numbers in an ascending sequence is most widely used.

</p>

<p>

Sorting Algoritms are widely used in many other tasks such as searching, merging, Data Parsing, etc. <br>

For eg:- <u>Binary Search</u> which is a well known searching algorithm that runs in O(logn) time needs the data to be sorted in

increasing order.

</p>

<p>

There are many different sorting algorithms, each with its own specific unique intuition and implementation.

They are classified according to two metrics: space complexity and time complexity.<br><br>

Those two kinds of complexity are represented with <a href="https://www.geeksforgeeks.org/asymptotic-analysis-comparison-sorting-algorithms/">asymptotic notations</a>,

mainly with the symbols O, <span class="symbol">Θ</span>, <span class="symbol">Ω</span>, representing

respectively the upper bound, the tight bound, and the lower bound of the algorithm's complexity, specifying in brackets an expression in terms of

<code><var>n</var></code>, the number of the elements of the data sequence.<br><br>

Most of them fall into following categories:

</p>

<ul>

<li>Logarithmic<br>

The complexity is proportional to the binary logarithm (i.e to the base 2) of <code><var>n</var></code>.<br>

An example of a logarithmic sorting algorithm is Quick sort, with space and time complexity O<code>(n × log n)</code>.<br>

</li>

<li>Quadratic<br>

The complexity is proportional to the square of <code><var>n</var></code>.<br>

An example of a quadratic sorting algorithm is Bubble sort, with a time complexity of O<code>(n<sup>2</sup>)</code>.<br>

</li>

<li>Linear<br>

The complexity approaches to linear time of <code><var>n</var></code>.<br>

An example of linear sorting algorithm is Counting Sort, with time complexity of O<code>(n+k)</code>.<br>

</li>

</ul>

<p>

Space and time complexity can also be further subdivided into 3 different cases: <a href="https://www.geeksforgeeks.org/worst-average-and-best-case-analysis-of-algorithms/">

best case, average case and worst case</a>.

</p>

<p>

Some Sorting algorithms are easy to understand while some cause havoc in one's mind. While it's not easy to learn everything and it’s not a good recommendation

to learn a piece of code, so here we are with a fun way of understanding the sorting algorithms with visual depiction of prominent ones. So, Let’s get into it....

</p>

<button id="sort-btn" class="ripple-sort-btn open">Let's Sort Away!</button>

</div>

</div>

</div>

<footer class="footer">

<div class="footer-container">

<div class="footer-content" style="margin-left:0;">

<span class="footer-title" style="font-family: 'Jura', Courier, monospace;">SORT VISUALIZER</span><br>

<center>

<br>

Rishabh Chopra<br>

</center>

</div>

<div class="footer-content">

<center><span class="footer-title">Contacts:</span></center>

<br>

&emsp;rishabh4087@rla.du.ac.in<br>

</div>

</div>

<div class="credits">

Got any reccomendation? <a href="Feedback/feedbackmain.html">It's Feedback Time</a>!

</div>

</footer>

</body>

</html>

**BUBBLE SORT -**

<!DOCTYPE html>

<html>

<head>

<title>iSort - The Sort Visualizer </title>\

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<meta name="language" content="English">

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

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

<link rel="stylesheet" type="text/css" href="../css/visualize.css">

<link rel= "stylesheet" type= "text/css" href= "../css/style2.css">

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

</head>

<body>

<script src="https://unpkg.com/aos@2.3.1/dist/aos.js"></script>

<script type="text/javascript" src="../javascript/index.js"></script>

<div class="sidenav" id="sidenav">

<a href="../index.html" class="sidenav-title no-remove"><i class="material-icons icon">home</i> Home</a>

<div id="sep" class="no-remove"></div>

<div class="sidenav-title no-remove"><i class="material-icons icon">sort</i> Sorts</div>

<a href="selectionSort.html" class="sidenav-element no-remove">Selection Sort</a>

<a href="insertionSort.html" class="sidenav-element no-remove">Insertion Sort</a>

<a href="countingSort.html" class="sidenav-element no-remove">Counting Sort</a>

</div>

<div class="topnav">

<button class="sidenav-btn topnav-element ripple open" id="sidenav-menu"><i class="material-icons icon">menu</i></button>

<a href="/" class="topnav-element">SORTING VISUALIZER - BUBBLE SORT</a>

</div>

<div class="screensplit">

<div class="hrsplit1">

<div class="vrsplit1"></div>

<div class="vrsplit2" style="overflow-y:scroll;">

<br>

<p id="line0">&nbsp BubbleSort(A[],n)</p>

<p id="line1">&nbsp &nbsp 1. &nbsp For i = 0 to n - 1</p>

<p id="line2">&nbsp &nbsp 2. &nbsp &nbsp For j = 0 to n - 1 - i</p>

<p id="line3">&nbsp &nbsp 3. &nbsp &nbsp &nbsp If A[j]>A[j+1]</p>

<p id="line4">&nbsp &nbsp 4. &nbsp &nbsp &nbsp &nbsp swap A[j] <-> A[j+1]</p>

<p id="line5">&nbsp &nbsp 5. &nbsp &nbsp &nbsp End If</p>

<p id="line6">&nbsp &nbsp 6. &nbsp &nbsp End For</p>

<p id="line7">&nbsp &nbsp 7. &nbsp End For</p>

<hr size="3" noshade>

<center>

<h4 style="color:aliceblue; margin: 0 0 0 0;" id="comp">No of Comparisons:</h4>

</center>

<hr size="3" noshade>

</div>

</div>

<div class="hrsplit2">

<div class="split1">

<div style="margin: auto; width: fit-content">

<center>

<label style="color:crimson; font-size: 20px;"><b>Number of bars :</b></label>

<input id="nele" type="number" min="1" max="400" value="25"></input>

&emsp;

<button class="btn1" type="button" onclick="generatebest()" id="Button4">

Generate Best Case Input

</button>

<button class="btn1" type="button" onclick="generateworst()" id="Button5">

Generate Worst Case Input

</button>

&emsp;

<label style="color:crimson; font-size: 20px;"><b>Speed :</b></label>

<input id="speeder" type="range" min="5" max="100" value="40" oninput="delaySet()"></input>

<br>

<button class="btn1" type="button" onclick="generate()" id="Button1">

Generate

</button>

&emsp;

<button class="btn1" type="button" onclick="generate2()" id="Button2">

Generate Random Array

</button>

&emsp;

<button class="btn2" onclick="delaySet(),BubbleSort(),disable()" id="Button3">

Bubble Sort

</button>

&emsp;&emsp;

<button class="btn2" id="pauseButton">

Pause

</button>

&emsp;&emsp;

<button class="btn2" id="terminateButton" onclick="terminate=true;">

Terminate

</button>

</center>

</div>

</div>

<div class="split2">

<center>

<table style="border-collapse: collapse;">

<tr>

<center>

<h4 style="margin-top: 0px; margin-bottom: 5px;">Complexity </h4>

</center>

</tr>

<tr>

<td colspan="3">

<center>Time</center>

</td>

<td>

Space

</td>

</tr>

<tr>

<td>Best Case</td>

<td>Average Case</td>

<td>Worst Case</td>

<td rowspan="2"> <center>O(1)</center> </td>

</tr>

<tr>

<td><center>O(n)</center></td>

<td><center>O(n<sup>2</sup>)</center></td>

<td><center>O(n<sup>2</sup>)</center></td>

</tr>

</table>

</center>

</div>

</div>

</div>

<script type="text/javascript" src="../javascript/bubbleSort.js"></script>

<div id="cover">

<div id="description-box">

<h2 style="font-family: 'Electrolize', Courier, monospace;">

Bubble Sort

</h2>

<div class="description-content">

<p>

<b>What is the working mechanism of <a href="https://www.geeksforgeeks.org/bubble-sort/">Bubble Sort</a>, and what is

the origin of its name?</b>

<br> <br>

The bubble sort algorithm gets its name from the way that bubbles rise to the surface of a liquid - larger bubbles rise

faster - larger elements "rise" to their correct positions faster in the list. During each pass through the list,

adjacent elements are compared and swapped if necessary, gradually "sorting" the elements as if they were rising to the surface

like bubbles.

</p>

<br>

<p>

<b>Is bubble sort a <a href="https://www.geeksforgeeks.org/stable-and-unstable-sorting-algorithms/">stable</a>,

<a href="https://www.geeksforgeeks.org/in-place-algorithm/">in-place</a> sorting algorithm or an out-of-place sorting algorithm?</b>

<br> <br>

Bubble sort is a stable in-place algorithm, meaning that it does not require any extra space and maintains the relative order of duplicates.

</p>

<br>

<p>

<b>How does the number of elements impact the <a href="https://www.khanacademy.org/computing/ap-computer-science-

principles/algorithms-101/evaluating-algorithms/a/measuring-an-algorithms-efficiency">efficiency</a> of Bubble sort?</b>

<br> <br>

The efficiency of Bubble sort is impacted by the size of the input list, and as the list size increases,

the number of iterations needed for the algorithm to sort the list also increases, which can result in poor

performance when sorting large lists, making it unsuitable for large-scale applications.

</p>

<br>

<p>

<b>What are advantages and disadvantages of Bubble Sort?</b>

<br> <br>

Advantages:

<ul>

<li>Bubble sort is easy to understand and implement.</li>

<li>It does not require any additional memory space.</li>

<li>It's adaptability to different types of data.</li>

<li>It is a stable sorting algorithm, meaning that elements with the same key value maintain their relative order in the sorted output.</li>

</ul>

<br>

Disadvantages:

<ul>

<li>Bubble sort has a time complexity of O(n^2) which makes it very slow for large data sets.</li>

<li>It is not efficient for large data sets, because it requires multiple passes through the data.</li>

</ul>

</p>

</div>

</div>

</div>

<footer class="footer">

<div class="credits">

Got any reccomendation? <a href="../Feedback/feedbackmain.html">It's Feedback Time</a>!

</div>

</footer>

</body>

</html>

**INSERTION SORT -**

<!DOCTYPE html>

<html>

<head>

<title>iSort - The Sort Visualizer </title>\

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<meta name="language" content="English">

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

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

<link rel="stylesheet" type="text/css" href="../css/visualize.css">

<link rel= "stylesheet" type= "text/css" href= "../css/style2.css">

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

</head>

<body>

<script src="https://unpkg.com/aos@2.3.1/dist/aos.js"></script>

<script type="text/javascript" src="../javascript/index.js"></script>

<div class="sidenav" id="sidenav">

<a href="../index.html" class="sidenav-title no-remove"><i class="material-icons icon">home</i> Home</a>

<div id="sep" class="no-remove"></div>

<div class="sidenav-title no-remove"><i class="material-icons icon">sort</i> Sorts</div>

<a href="selectionSort.html" class="sidenav-element no-remove">Selection Sort</a>

<a href="bubbleSort.html" class="sidenav-element no-remove">Bubble Sort</a>

<a href="countingSort.html" class="sidenav-element no-remove">Counting Sort</a>

</div>

<div class="topnav">

<button class="sidenav-btn topnav-element ripple open" id="sidenav-menu"><i class="material-icons icon">menu</i></button>

<a href="/" class="topnav-element">SORTING VISUALIZER - INSERTION SORT</a>

</div>

<div class="screensplit">

<div class="hrsplit1">

<div class="vrsplit1"></div>

<div class="vrsplit2" style="overflow-y:scroll;">

<br>

<p id="line0">&nbsp InsertionSort(A[],n)</p>

<p id="line1">&nbsp &nbsp 1. &nbsp For i = 1 to n</p>

<p id="line2">&nbsp &nbsp 2. &nbsp &nbsp Key<-Arr[i]</p>

<p id="line3">&nbsp &nbsp 3. &nbsp &nbsp j<-[i-1]</p>

<p id="line4">&nbsp &nbsp 4. &nbsp &nbsp while(j>0 and Arr[j]>Key)</p>

<p id="line5">&nbsp &nbsp 5. &nbsp &nbsp &nbsp &nbsp A[j+1]<-A[j]</p>

<p id="line6">&nbsp &nbsp 6. &nbsp &nbsp &nbsp &nbsp j<-j-1</p>

<p id="line7">&nbsp &nbsp 7. &nbsp &nbsp A[j+1]<-Key</p>

<hr size="3" noshade>

<center>

<h4 style="color:aliceblue; margin: 0 0 0 0;" id="comp">No of Comparisons:</h4>

</center>

<hr size="3" noshade>

</div>

</div>

<div class="hrsplit2">

<div class="split1">

<div style="margin: auto; width: fit-content">

<center>

<label style="color:crimson; font-size: 20px;"><b>Number of bars :</b></label>

<input id="nele" type="number" min="1" max="400" value="25"></input>

&emsp;

<button class="btn1" type="button" onclick="generatebest()" id="Button4">

Generate Best Case Input

</button>

<button class="btn1" type="button" onclick="generateworst()" id="Button5">

Generate Worst Case Input

</button>

&emsp;

<label style="color:crimson; font-size: 20px;"><b>Speed :</b></label>

<input id="speeder" type="range" min="5" max="100" value="40" oninput="delaySet()"></input>

<br>

<button class="btn1" type="button" onclick="generate()" id="Button1">

Generate

</button>

&emsp;

<button class="btn1" type="button" onclick="generate2()" id="Button2">

Generate Random Array

</button>

&emsp;

<button class="btn2" onclick="delaySet(),InsertionSort(),disable()" id="Button3">

Insertion Sort

</button>

&emsp;&emsp;

<button class="btn2" id="pauseButton">

Pause

</button>

&emsp;&emsp;

<button class="btn2" id="terminateButton" onclick="terminate=true;">

Terminate

</button>

</center>

</div>

</div>

<div class="split2">

<center>

<table style="border-collapse: collapse;">

<tr>

<center>

<h4 style="margin-top: 0px; margin-bottom: 5px;">Complexity </h4>

</center>

</tr>

<tr>

<td colspan="3">

<center>Time</center>

</td>

<td>

Space

</td>

</tr>

<tr>

<td>Best Case</td>

<td>Average Case</td>

<td>Worst Case</td>

<td rowspan="2"> <center>O(1)</center> </td>

</tr>

<tr>

<td><center>O(n)</center></td>

<td><center>O(n<sup>2</sup>)</center></td>

<td><center>O(n<sup>2</sup>)</center></td>

</tr>

</table>

</center>

</div>

</div>

</div>

<script type="text/javascript" src="../javascript/insertionSort.js"></script>

<div id="cover">

<div id="description-box">

<h2 style="font-family: 'Electrolize', Courier, monospace;">

Insertion Sort

</h2>

<div class="description-content">

<p>

<b>What is the working mechanism of <a href="https://www.geeksforgeeks.org/insertion-sort/">Insertion Sort</a>, and what is

the origin of its name?</b>

<br> <br>

Insertion sort is a simple sorting algorithm that works

similar to the way you sort playing cards in your hands.

The array is virtually split into a sorted and an unsorted part.

Values from the unsorted part are picked and placed at the correct

position in the sorted part.

</p>

<br>

<p>

<b>Is insertion sort a <a href="https://www.geeksforgeeks.org/stable-and-unstable-sorting-algorithms/">stable</a>,

<a href="https://www.geeksforgeeks.org/in-place-algorithm/">in-place</a> sorting algorithm or an out-of-place sorting algorithm?</b>

<br> <br>

Insertion sort is a stable in-place algorithm, meaning that it does not require any extra space and maintains the relative order of duplicates.

</p>

<br>

<p>

<b>How does the number of elements impact the <a href="https://www.khanacademy.org/computing/ap-computer-science-

principles/algorithms-101/evaluating-algorithms/a/measuring-an-algorithms-efficiency">efficiency</a> of insertion sort?</b>

<br> <br>

The efficiency of insertion sort is impacted by the size of the input list, and as the list size increases,

the number of iterations needed for the algorithm to sort the list also increases, which can result in poor

performance when sorting large lists, making it unsuitable for large-scale applications.

</p>

<br>

<p>

<b>What are advantages and disadvantages of insertion Sort?</b>

<br> <br>

Advantages:

<ul>

<li>Insertion sort is easy to understand and implement.</li>

<li>It does not require any additional memory space.</li>

<li>It's adaptability to different types of data.</li>

<li>It is a stable sorting algorithm, meaning that elements with the same key value maintain their relative order in the sorted output.</li>

</ul>

<br>

Disadvantages:

<ul>

<li>Insertion sort has a time complexity of O(n^2) which makes it very slow for large data sets.</li>

<li>It is not efficient for large data sets, because it requires multiple passes through the data.</li>

</ul>

</p>

</div>

</div>

</div>

<footer class="footer">

<div class="credits">

Got any reccomendation? <a href="../Feedback/feedbackmain.html">It's Feedback Time!</a>

</div>

</footer>

</body>

</html>

**SELECTION SORT -**

<!DOCTYPE html>

<html>

<head>

<title>iSort - The Sort Visualizer </title>\

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<meta name="language" content="English">

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

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

<link rel="stylesheet" type="text/css" href="../css/visualize.css">

<link rel= "stylesheet" type= "text/css" href= "../css/style2.css">

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

</head>

<body>

<script src="https://unpkg.com/aos@2.3.1/dist/aos.js"></script>

<script type="text/javascript" src="../javascript/index.js"></script>

<div class="sidenav" id="sidenav">

<a href="../index.html" class="sidenav-title no-remove"><i class="material-icons icon">home</i> Home</a>

<div id="sep" class="no-remove"></div>

<div class="sidenav-title no-remove"><i class="material-icons icon">sort</i> Sorts</div>

<a href="bubbleSort.html" class="sidenav-element no-remove">Bubble Sort</a>

<a href="insertionSort.html" class="sidenav-element no-remove">Insertion Sort</a>

<a href="countingSort.html" class="sidenav-element no-remove">Counting Sort</a>

</div>

<div class="topnav">

<button class="sidenav-btn topnav-element ripple open" id="sidenav-menu"><i class="material-icons icon">menu</i></button>

<a href="/" class="topnav-element">SORTING VISUALIZER - SELECTION SORT</a>

</div>

<div class="screensplit">

<div class="hrsplit1">

<div class="vrsplit1"></div>

<div class="vrsplit2" style="overflow-y:scroll;">

<br>

<p id="line0">&nbsp SelectionSort(A[],n)</p>

<p id="line1">&nbsp &nbsp 1. &nbsp For i = 0 to n - 2</p>

<p id="line2">&nbsp &nbsp 2. &nbsp &nbsp min =i</p>

<p id="line3">&nbsp &nbsp 3. &nbsp &nbsp For j =i+1 to n-1</p>

<p id="line4">&nbsp &nbsp 4. &nbsp &nbsp &nbsp If A[j] < A[min])</p>

<p id="line5">&nbsp &nbsp 5. &nbsp &nbsp &nbsp &nbsp min = j</p>

<p id="line6">&nbsp &nbsp 6. &nbsp &nbsp End For</p>

<p id="line7">&nbsp &nbsp 7. &nbsp &nbsp swap A[min] <-> A[i];</p>

<hr size="3" noshade>

<center>

<h4 style="color:aliceblue; margin: 0 0 0 0;" id="comp">No of Comparisons:</h4>

</center>

<hr size="3" noshade>

</div>

</div>

<div class="hrsplit2">

<div class="split1">

<div style="margin: auto; width: fit-content">

<center>

<label style="color:crimson; font-size: 20px;"><b>Number of bars :</b></label>

<input id="nele" type="number" min="1" max="400" value="25"></input>

&emsp;

<button class="btn1" type="button" onclick="generatebest()" id="Button4">

Generate Best Case Input

</button>

<button class="btn1" type="button" onclick="generateworst()" id="Button5">

Generate Worst Case Input

</button>

&emsp;

<label style="color:crimson; font-size: 20px;"><b>Speed :</b></label>

<input id="speeder" type="range" min="5" max="100" value="40" oninput="delaySet()"></input>

<br>

<button class="btn1" type="button" onclick="generate()" id="Button1">

Generate

</button>

&emsp;

<button class="btn1" type="button" onclick="generate2()" id="Button2">

Generate Random Array

</button>

&emsp;

<button class="btn2" onclick="delaySet(),selectionSort(),disable()" id="Button3">

Selection Sort

</button>

&emsp;&emsp;

<button class="btn2" id="pauseButton">

Pause

</button>

&emsp;&emsp;

<button class="btn2" id="terminateButton" onclick="terminate=true;">

Terminate

</button>

</center>

</div>

</div>

<div class="split2">

<center>

<table style="border-collapse: collapse;">

<tr>

<center>

<h4 style="margin-top: 0px; margin-bottom: 5px;">Complexity </h4>

</center>

</tr>

<tr>

<td colspan="3">

<center>Time</center>

</td>

<td>

Space

</td>

</tr>

<tr>

<td>Best Case</td>

<td>Average Case</td>

<td>Worst Case</td>

<td rowspan="2"> <center>O(1)</center> </td>

</tr>

<tr>

<td><center>O(n)</center></td>

<td><center>O(n<sup>2</sup>)</center></td>

<td><center>O(n<sup>2</sup>)</center></td>

</tr>

</table>

</center>

</div>

</div>

</div>

<script type="text/javascript" src="../javascript/selectionSort.js"></script>

<div id="cover">

<div id="description-box">

<h2 style="font-family: 'Electrolize', Courier, monospace;">

Selection Sort

</h2>

<div class="description-content">

<p>

<b>What is the working mechanism of <a href="https://www.geeksforgeeks.org/selection-sort/">Selection Sort</a> ?</b>

<br> <br>

Selection sort is a simple sorting algorithm that sorts an array by repeatedly finding the minimum element from unsorted

part of the array and placing it at the beginning of the sorted part of the array.

</p>

<br>

<p>

<b>Is selection sort a <a href="https://www.geeksforgeeks.org/stable-and-unstable-sorting-algorithms/">stable</a>,

<a href="https://www.geeksforgeeks.org/in-place-algorithm/">in-place</a> sorting algorithm or an out-of-place sorting algorithm?</b>

<br> <br>

Selection sort is an unstable in-place algorithm, meaning that it does not require any extra space but can't maintain the relative order of duplicates.

</p>

<br>

<p>

<b>How does the number of elements impact the <a href="https://www.khanacademy.org/computing/ap-computer-science-

principles/algorithms-101/evaluating-algorithms/a/measuring-an-algorithms-efficiency">efficiency</a> of Selection sort?</b>

<br> <br>

The time complexity of selection sort is O(n^2), where n is the number of elements

in the array. This means that the efficiency of selection sort is directly impacted by the number of elements in the array.

As the number of elements increases, the time taken to sort the array using selection sort also increases.

</p>

<br>

<p>

<b>What are advantages and disadvantages of Selection Sort?</b>

<br> <br>

Advantages:

<ul>

<li>Works well with small datasets.</li>

<li>It does not require any additional memory space.</li>

<li>It's adaptability to different types of data.</li>

<li>It is easy to modify to sort in ascending or descending order.</li>

<br>

Disadvantages:

<ul>

<li>Selection sort has a time complexity of O(n^2) in the worst and average case.</li>

<li>Not adaptive, meaning it doesn’t take advantage of the fact that the list may already be sorted or partially sorted</li>

</ul>

</p>

</div>

</div>

</div>

<footer class="footer">

<div class="credits">

Got any reccomendation? <a href="../Feedback/feedbackmain.html">It's Feedback Time</a>!

</div>

</footer>

</body>

</html>

**COUNTING SORT -**

<!DOCTYPE html>

<html>

<head>

<title>iSort - The Sort Visualizer </title>\

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<meta name="language" content="English">

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

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

<link rel="stylesheet" type="text/css" href="../css/visualize\_linear.css">

<link rel= "stylesheet" type= "text/css" href= "../css/style2.css">

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

</head>

<body>

<script src="https://unpkg.com/aos@2.3.1/dist/aos.js"></script>

<script type="text/javascript" src="../javascript/index.js"></script>

<div class="sidenav" id="sidenav">

<a href="../index.html" class="sidenav-title no-remove"><i class="material-icons icon">home</i> Home</a>

<div id="sep" class="no-remove"></div>

<div class="sidenav-title no-remove"><i class="material-icons icon">sort</i> Sorts</div>

<a href="bubbleSort.html" class="sidenav-element no-remove">Bubble Sort</a>

<a href="selectionSort.html" class="sidenav-element no-remove">Selection Sort</a>

<a href="insertionSort.html" class="sidenav-element no-remove">Insertion Sort</a>

</div>

<div class="topnav">

<button class="sidenav-btn topnav-element ripple open" id="sidenav-menu"><i class="material-icons icon">menu</i></button>

<a href="/" class="topnav-element">SORTING VISUALIZER - COUNTING SORT</a>

</div>

<div class="screensplit">

<div class="hrsplit1">

<div class="vrsplit1">

<div class="mainarray"></div>

<div class="countarray"></div>

</div>

<div class="vrsplit2" style="overflow-y:scroll;">

<br>

<p id="line0">&nbsp CountingSort(A[],n)</p>

<p id="line1">&nbsp &nbsp 1. &nbsp Initialize count array C[k] <br>&nbsp &nbsp &nbsp &nbsp where k=maxele(A)</p>

<p id="line2">&nbsp &nbsp 2. &nbsp For j = 0 to n - 1</p>

<p id="line3">&nbsp &nbsp 3. &nbsp &nbsp do C[A[j]]=C[A[j]]+1</p>

<p id="line4">&nbsp &nbsp 4. &nbsp For j = 1 to k</p>

<p id="line5">&nbsp &nbsp 5. &nbsp &nbsp do C[j] += C[j-1]</p>

<p id="line6">&nbsp &nbsp 6. &nbsp For j = n-1 to 0</p>

<p id="line7">&nbsp &nbsp 7. &nbsp &nbsp B[C[A[j]]-1] = A[j]</p>

<p id="line8">&nbsp &nbsp 8. &nbsp &nbsp C[A[j]] -= 1</p>

<hr size="3" noshade>

</div>

</div>

<div class="hrsplit2">

<div class="split1">

<div style="margin: auto; width: fit-content">

<center>

<label style="color:crimson; font-size: 20px;"><b>Number of bars :</b></label>

<input id="nele" type="number" min="1" max="400" value="25"></input>

&emsp;

<button class="btn1" type="button" onclick="generatebest()" id="Button4">

Generate Best Case Input

</button>

<button class="btn1" type="button" onclick="generate2()" id="Button5">

Generate Worst Case Input

</button>

&emsp;

<label style="color:crimson; font-size: 20px;"><b>Speed :</b></label>

<input id="speeder" type="range" min="5" max="100" value="40" oninput="delaySet()"></input>

<br>

<button class="btn1" type="button" onclick="generate()" id="Button1">

Generate

</button>

&emsp;

<button class="btn1" type="button" onclick="generate2()" id="Button2">

Generate Random Array

</button>

&emsp;

<button class="btn2" onclick="delaySet(),CountingSort(),disable()" id="Button3">

Counting Sort

</button>

&emsp;&emsp;

<button class="btn2" id="pauseButton">

Pause

</button>

&emsp;&emsp;

<button class="btn2" id="terminateButton" onclick="terminate=true;">

Terminate

</button>

</center>

</div>

</div>

<div class="split2">

<center>

<table style="border-collapse: collapse;">

<tr>

<center>

<h4 style="margin-top: 0px; margin-bottom: 5px;">Complexity </h4>

</center>

</tr>

<tr>

<td colspan="3">

<center>Time</center>

</td>

<td>

Space

</td>

</tr>

<tr>

<td>Best Case</td>

<td>Average Case</td>

<td>Worst Case</td>

<td rowspan="2"> <center>O(k)</center> </td>

</tr>

<tr>

<td><center>O(n+k)</center></td>

<td><center>O(n+k)</center></td>

<td><center>O(n+k)</center></td>

</tr>

</table>

</center>

</div>

</div>

</div>

<script type="text/javascript" src="../javascript/countingSort.js"></script>

<div id="cover">

<div id="description-box">

<h2 style="font-family: 'Electrolize', Courier, monospace;">

Counting Sort

</h2>

<div class="description-content">

<p>

<b>What is the working mechanism of <a href="https://www.geeksforgeeks.org/counting-sort/">Counting Sort</a>, and what is

the origin of its name?</b>

<br> <br>

Counting sort is a sorting technique based on keys between a specific range.

It works by counting the number of objects having distinct key values (a kind of hashing).

Then do some arithmetic operations to calculate the position of each object in the output sequence.

</p>

<br>

<p>

<b>Is counting sort a <a href="https://www.geeksforgeeks.org/stable-and-unstable-sorting-algorithms/">stable</a>,

<a href="https://www.geeksforgeeks.org/in-place-algorithm/">in-place</a> sorting algorithm or an out-of-place sorting algorithm?</b>

<br> <br>

Counting sort is a stable out-of-place algorithm, meaning that it require extra space and maintains the relative order of duplicates.

</p>

<br>

<p>

<b>How does the number of elements impact the <a href="https://www.khanacademy.org/computing/ap-computer-science-

principles/algorithms-101/evaluating-algorithms/a/measuring-an-algorithms-efficiency">efficiency</a> of Counting sort?</b>

<br> <br>

Counting sort algorithm work best if k is not significantly larger than n.

In this case the complexity becomes close to O(n) or linear.

</p>

<p>

<b>What are advantages and disadvantages of Counting Sort?</b>

<br> <br>

Advantages:

<ul>

<li>Counting sort generally performs faster than all comparison-based sorting algorithms,

such as merge sort and quicksort, if the range of input is of the order of the number of input.</li>

<li>Counting sort is easy to code.</li>

</ul>

<br>

Disadvantages:

<ul>

<li>Counting sort doesn't work on decimal values.</li>

<li>Counting sort is inefficient if the range of values to be sorted is very large.</li>

</ul>

</p>

</div>

</div>

</div>

<footer class="footer">

<div class="credits">

Got any reccomendation? <a href="../Feedback/feedbackmain.html">It's Feedback Time</a>!

</div>

</footer>

</body>

</html>

**JAVA SCRIPT FILES-**

**INDEX FILE -**

AOS.init({

duration: 1000,

once: true

});

const byId = id => {

return document.getElementById(id);

}

window.onclick = function(event) {

if (event.target.matches(".open")) {

byId("sidenav").classList.toggle("show");

} else if (!(event.target.matches(".no-remove"))) {

byId("sidenav").classList.remove("show")

}

}

**BUBBLE SORT -**

const container = document.querySelector(".vrsplit1");

let isPlaying = false;

const pausePlayBtn = document.getElementById("pauseButton");

let terminate = false;

var count = 0;

const compare = document.getElementById("comp");

function generatebars(num) {

container.innerHTML=""

for (let i = 0; i < num; i += 1) {

const value = Math.floor(Math.random() \* (180-9)+9) + 1;

const bar = document.createElement("div");

bar.classList.add("bar");

bar.style.height = `${value/2}%`;

const barLabel = document.createElement("label");

barLabel.classList.add("bar\_\_id");

barLabel.innerHTML = value;

if (num>80) {

barLabel.style.display='none';

}

if (num<=40) {

if (num<=10) {

barLabel.style.fontSize = 'xxx-large';

}

else if (num<=20) {

barLabel.style.fontSize = 'xx-large';

}

if (num>20 && num<=30) {

barLabel.style.fontSize = 'x-large';

}

else if (num<=40) {

barLabel.style.fontSize = 'large';

}

}

bar.appendChild(barLabel);

container.appendChild(bar);

}

}

generatebars(25);

function generate() {

var n = document.getElementById("nele");

var numele = parseInt(n.value);

if (numele>400) {

window.alert("Upper bound is 400 bars. Kindly choose a value in that range!");

n.value=400;

generate();

}

else {

generatebars(numele);

}

}

function generate2() {

const value = Math.floor(Math.random() \* 80) + 1;

generatebars(value);

}

function generatebarsWorst(num) {

container.innerHTML=""

let values = Array.from({length: num}, (\_, i) => Math.max(num - i + 9, 10));

for (let i = 0; i < num; i += 1) {

const value = values[i];

const bar = document.createElement("div");

bar.classList.add("bar");

bar.style.height = `${value/2}%`;

const barLabel = document.createElement("label");

barLabel.classList.add("bar\_\_id");

barLabel.innerHTML = value;

if (num>80) {

barLabel.style.display='none';

}

if (num<=40) {

if (num<=10) {

barLabel.style.fontSize = 'xxx-large';

}

else if (num<=20) {

barLabel.style.fontSize = 'xx-large';

}

if (num>20 && num<=30) {

barLabel.style.fontSize = 'x-large';

}

else if (num<=40) {

barLabel.style.fontSize = 'large';

}

}

bar.appendChild(barLabel);

container.appendChild(bar);

}

}

function generatebarsBest(num) {

container.innerHTML=""

let values = Array.from({length: num}, (\_, i) => i + 1 + 9);

for (let i = 0; i < num; i += 1) {

const value = values[i];

const bar = document.createElement("div");

bar.classList.add("bar");

bar.style.height = `${value/2}%`;

const barLabel = document.createElement("label");

barLabel.classList.add("bar\_\_id");

barLabel.innerHTML = value;

if (num>80) {

barLabel.style.display='none';

}

if (num<=40) {

if (num<=10) {

barLabel.style.fontSize = 'xxx-large';

}

else if (num<=20) {

barLabel.style.fontSize = 'xx-large';

}

if (num>20 && num<=30) {

barLabel.style.fontSize = 'x-large';

}

else if (num<=40) {

barLabel.style.fontSize = 'large';

}

}

bar.appendChild(barLabel);

container.appendChild(bar);

}

}

function generatebest() {

const value = Math.floor(Math.random() \* 80) + 1;

generatebarsBest(value);

}

function generateworst() {

const value = Math.floor(Math.random() \* 80) + 1;

generatebarsWorst(value);

}

function disable() {

document.getElementById("Button1").disabled = true;

document.getElementById("Button1").style.backgroundColor = "#d8b6ff";

document.getElementById("Button2").disabled = true;

document.getElementById("Button2").style.backgroundColor = "#d8b6ff";

document.getElementById("Button3").disabled = true;

document.getElementById("Button3").style.backgroundColor = "#d8b6ff";

document.getElementById("Button4").disabled = true;

document.getElementById("Button4").style.backgroundColor = "#d8b6ff";

document.getElementById("Button5").disabled = true;

document.getElementById("Button5").style.backgroundColor = "#d8b6ff";

}

var delay = 5000;

async function BubbleSort() {

count = 0;

let bars = document.querySelectorAll(".bar");

var l0 = document.getElementById("line0");

l0.style.backgroundColor = "lightgreen";

for (let i = 0; i < bars.length; i++) {

var l1 = document.getElementById("line1");

l1.style.backgroundColor = "cyan";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

l1.style.backgroundColor = null;

var c2 = 1;

for (let j = 0; j < bars.length - i - 1; j++) {

var l2 = document.getElementById("line2");

l2.style.backgroundColor = "cyan";

while (isPlaying) {

if (terminate) {

l2.style.backgroundColor = null;

l0.style.backgroundColor = null;

document.getElementById("Button1").disabled = false;

document.getElementById("Button1").style.backgroundColor = "#a54997";

document.getElementById("Button2").disabled = false;

document.getElementById("Button2").style.backgroundColor = "#a54997";

document.getElementById("Button3").disabled = false;

document.getElementById("Button3").style.backgroundColor = "#a54997";

document.getElementById("Button4").disabled = false;

document.getElementById("Button4").style.backgroundColor = "#a54997";

document.getElementById("Button5").disabled = false;

document.getElementById("Button5").style.backgroundColor = "#a54997";

isPlaying = false;

pausePlayBtn.textContent = 'Pause';

compare.textContent=' ' + " No of Comparisons: ";

for (let k=0;k<bars.length;k++) {

bars[k].style.backgroundColor="rgb(236, 190, 53)";

}

terminate = !terminate;

return;

}

await new Promise((resolve) => setTimeout(() => { resolve(); }, 1000));

}

if (terminate) {

l2.style.backgroundColor = null;

l0.style.backgroundColor = null;

document.getElementById("Button1").disabled = false;

document.getElementById("Button1").style.backgroundColor = "#a54997";

document.getElementById("Button2").disabled = false;

document.getElementById("Button2").style.backgroundColor = "#a54997";

document.getElementById("Button3").disabled = false;

document.getElementById("Button3").style.backgroundColor = "#a54997";

document.getElementById("Button4").disabled = false;

document.getElementById("Button4").style.backgroundColor = "#a54997";

document.getElementById("Button5").disabled = false;

document.getElementById("Button5").style.backgroundColor = "#a54997";

for (let k=0;k<bars.length;k++) {

bars[k].style.backgroundColor="rgb(236, 190, 53)";

}

compare.textContent=' ' + " No of Comparisons: ";

terminate = !terminate;

return;

}

var value1 = parseInt(bars[j].childNodes[0].innerHTML);

var value2 = parseInt(bars[j + 1].childNodes[0].innerHTML);

var l3 = document.getElementById("line3");

l3.style.backgroundColor = "cyan";

if (value1 > value2) {

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

var l4 = document.getElementById("line4");

l4.style.backgroundColor = "cyan";

bars[j].style.backgroundColor = "red";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

bars[j + 1].style.backgroundColor = "red";

var temp1 = bars[j].style.height;

var temp2 = bars[j].childNodes[0].innerText;

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

bars[j].style.height = bars[j + 1].style.height;

bars[j].childNodes[0].innerText = bars[j + 1].childNodes[0].innerText;

bars[j + 1].style.height = temp1;

bars[j + 1].childNodes[0].innerText = temp2;

l4.style.backgroundColor = null;

}

c2 = c2+1;

var l5 = document.getElementById("line5");

l5.style.backgroundColor = "cyan";

l3.style.backgroundColor = null;

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

l5.style.backgroundColor = null;

bars[j].style.backgroundColor = "rgb(236, 190, 53)";

bars[j + 1].style.backgroundColor = "rgb(236, 190, 53)";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

l2.style.backgroundColor = null;

}

count = count + c2;

compare.textContent=' ' + " No of Comparisons: "+count;

var l6 = document.getElementById("line6");

l6.style.backgroundColor = "cyan";

bars[bars.length - i - 1].style.backgroundColor = "rgb(49, 226, 13)";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

l6.style.backgroundColor = null;

}

var l7 = document.getElementById("line7");

l7.style.backgroundColor = "cyan";

for (let i = 0; i < bars.length; i++) {

bars[i].style.backgroundColor = "rgb(49, 226, 13)";

}

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

l7.style.backgroundColor = null;

document.getElementById("Button1").disabled = false;

document.getElementById("Button1").style.backgroundColor = "#a54997";

document.getElementById("Button2").disabled = false;

document.getElementById("Button2").style.backgroundColor = "#a54997";

document.getElementById("Button3").disabled = false;

document.getElementById("Button3").style.backgroundColor = "#a54997";

document.getElementById("Button4").disabled = false;

document.getElementById("Button4").style.backgroundColor = "#a54997";

document.getElementById("Button5").disabled = false;

document.getElementById("Button5").style.backgroundColor = "#a54997";

compare.textContent=' ' + " No of Comparisons: "+count;

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

l0.style.backgroundColor = null;

}

function delaySet() {

delay = 5000;

var s = document.getElementById("speeder");

var d = parseInt(s.value);

delay=delay/d;

}

pausePlayBtn.addEventListener('click', () => {

if (isPlaying) {

isPlaying = false;

pausePlayBtn.textContent = 'Pause';

} else {

isPlaying = true;

pausePlayBtn.textContent = 'Resume';

}

});

**INSERTION SORT -**

const container = document.querySelector(".vrsplit1");

let isPlaying = false;

const pausePlayBtn = document.getElementById("pauseButton");

let terminate = false;

var count = 0;

const compare = document.getElementById("comp");

function generatebars(num) {

container.innerHTML=""

for (let i = 0; i < num; i += 1) {

const value = Math.floor(Math.random() \* (180-9)+9) + 1;

const bar = document.createElement("div");

bar.classList.add("bar");

bar.style.height = `${value/2}%`;

const barLabel = document.createElement("label");

barLabel.classList.add("bar\_\_id");

barLabel.innerHTML = value;

if (num>80) {

barLabel.style.display='none';

}

if (num<=40) {

if (num<=10) {

barLabel.style.fontSize = 'xxx-large';

}

else if (num<=20) {

barLabel.style.fontSize = 'xx-large';

}

if (num>20 && num<=30) {

barLabel.style.fontSize = 'x-large';

}

else if (num<=40) {

barLabel.style.fontSize = 'large';

}

}

bar.appendChild(barLabel);

container.appendChild(bar);

}

}

generatebars(25);

function generate() {

var n = document.getElementById("nele");

var numele = parseInt(n.value);

if (numele>400) {

window.alert("Upper bound is 400 bars. Kindly choose a value in that range!");

n.value=400;

generate();

}

else {

generatebars(numele);

}

}

function generate2() {

const value = Math.floor(Math.random() \* 80) + 1;

generatebars(value);

}

function generatebarsWorst(num) {

container.innerHTML=""

let values = Array.from({length: num}, (\_, i) => Math.max(num - i + 9, 10));

for (let i = 0; i < num; i += 1) {

const value = values[i];

const bar = document.createElement("div");

bar.classList.add("bar");

bar.style.height = `${value/2}%`;

const barLabel = document.createElement("label");

barLabel.classList.add("bar\_\_id");

barLabel.innerHTML = value;

if (num>80) {

barLabel.style.display='none';

}

if (num<=40) {

if (num<=10) {

barLabel.style.fontSize = 'xxx-large';

}

else if (num<=20) {

barLabel.style.fontSize = 'xx-large';

}

if (num>20 && num<=30) {

barLabel.style.fontSize = 'x-large';

}

else if (num<=40) {

barLabel.style.fontSize = 'large';

}

}

bar.appendChild(barLabel);

container.appendChild(bar);

}

}

function generatebarsBest(num) {

container.innerHTML=""

let values = Array.from({length: num}, (\_, i) => i + 1 + 9);

for (let i = 0; i < num; i += 1) {

const value = values[i];

const bar = document.createElement("div");

bar.classList.add("bar");

bar.style.height = `${value/2}%`;

const barLabel = document.createElement("label");

barLabel.classList.add("bar\_\_id");

barLabel.innerHTML = value;

if (num>80) {

barLabel.style.display='none';

}

if (num<=40) {

if (num<=10) {

barLabel.style.fontSize = 'xxx-large';

}

else if (num<=20) {

barLabel.style.fontSize = 'xx-large';

}

if (num>20 && num<=30) {

barLabel.style.fontSize = 'x-large';

}

else if (num<=40) {

barLabel.style.fontSize = 'large';

}

}

bar.appendChild(barLabel);

container.appendChild(bar);

}

}

function generatebest() {

const value = Math.floor(Math.random() \* 80) + 1;

generatebarsBest(value);

}

function generateworst() {

const value = Math.floor(Math.random() \* 80) + 1;

generatebarsWorst(value);

}

function disable() {

document.getElementById("Button1").disabled = true;

document.getElementById("Button1").style.backgroundColor = "#d8b6ff";

document.getElementById("Button2").disabled = true;

document.getElementById("Button2").style.backgroundColor = "#d8b6ff";

document.getElementById("Button3").disabled = true;

document.getElementById("Button3").style.backgroundColor = "#d8b6ff";

document.getElementById("Button4").disabled = true;

document.getElementById("Button4").style.backgroundColor = "#d8b6ff";

document.getElementById("Button5").disabled = true;

document.getElementById("Button5").style.backgroundColor = "#d8b6ff";

}

var delay = 5000;

async function InsertionSort()

{

count = 1;

let bars = document.querySelectorAll(".bar");

var l0 = document.getElementById("line0");

var l1 = document.getElementById("line1");

var l2 = document.getElementById("line2");

var l3 = document.getElementById("line3");

var l4 = document.getElementById("line4");

var l5 = document.getElementById("line5");

var l6 = document.getElementById("line6");

var l7 = document.getElementById("line7");

l0.style.backgroundColor = "lightgreen";

for (let i = 1; i < bars.length; i++)

{

l1.style.backgroundColor = "cyan";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

count++;

while (isPlaying) {

if (terminate) {

l1.style.backgroundColor = null;

l4.style.backgroundColor = null;

document.getElementById("Button1").disabled = false;

document.getElementById("Button1").style.backgroundColor = "#a54997";

document.getElementById("Button2").disabled = false;

document.getElementById("Button2").style.backgroundColor = "#a54997";

document.getElementById("Button3").disabled = false;

document.getElementById("Button3").style.backgroundColor = "#a54997";

document.getElementById("Button4").disabled = false;

document.getElementById("Button4").style.backgroundColor = "#a54997";

document.getElementById("Button5").disabled = false;

document.getElementById("Button5").style.backgroundColor = "#a54997";

isPlaying = false;

pausePlayBtn.textContent = 'Pause';

compare.textContent=' ' + " No of Comparisons: ";

for (let k=0;k<bars.length;k++) {

bars[k].style.backgroundColor="rgb(236, 190, 53)";

}

terminate = !terminate;

return;

}

await new Promise((resolve) => setTimeout(() => { resolve(); }, 1000));

}

if (terminate) {

l1.style.backgroundColor = null;

l4.style.backgroundColor = null;

document.getElementById("Button1").disabled = false;

document.getElementById("Button1").style.backgroundColor = "#a54997";

document.getElementById("Button2").disabled = false;

document.getElementById("Button2").style.backgroundColor = "#a54997";

document.getElementById("Button3").disabled = false;

document.getElementById("Button3").style.backgroundColor = "#a54997";

document.getElementById("Button4").disabled = false;

document.getElementById("Button4").style.backgroundColor = "#a54997";

document.getElementById("Button5").disabled = false;

document.getElementById("Button5").style.backgroundColor = "#a54997";

for (let k=0;k<bars.length;k++) {

bars[k].style.backgroundColor="rgb(236, 190, 53)";

}

compare.textContent=' ' + " No of Comparisons: ";

terminate = !terminate;

return;

}

var key = parseInt(bars[i].childNodes[0].innerHTML);

bars[i].style.backgroundColor = "White";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

l2.style.backgroundColor = "cyan";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

l2.style.backgroundColor = null;

var h = bars[i].style.height;

var j = i-1;

l3.style.backgroundColor = "cyan";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

l3.style.backgroundColor = null;

while (j>=0 && parseInt(bars[j].childNodes[0].innerHTML)>key)

{

l4.style.backgroundColor = "cyan";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

count++;

while (isPlaying) {

if (terminate) {

l1.style.backgroundColor = null;

l4.style.backgroundColor = null;

document.getElementById("Button1").disabled = false;

document.getElementById("Button1").style.backgroundColor = "#a54997";

document.getElementById("Button2").disabled = false;

document.getElementById("Button2").style.backgroundColor = "#a54997";

document.getElementById("Button3").disabled = false;

document.getElementById("Button3").style.backgroundColor = "#a54997";

document.getElementById("Button4").disabled = false;

document.getElementById("Button4").style.backgroundColor = "#a54997";

document.getElementById("Button5").disabled = false;

document.getElementById("Button5").style.backgroundColor = "#a54997";

isPlaying = false;

pausePlayBtn.textContent = 'Pause';

compare.textContent=' ' + " No of Comparisons: ";

for (let k=0;k<bars.length;k++) {

bars[k].style.backgroundColor="rgb(236, 190, 53)";

}

terminate = !terminate;

return;

}

await new Promise((resolve) => setTimeout(() => { resolve(); }, 1000));

}

if (terminate) {

l1.style.backgroundColor = null;

l4.style.backgroundColor = null;

document.getElementById("Button1").disabled = false;

document.getElementById("Button1").style.backgroundColor = "#a54997";

document.getElementById("Button2").disabled = false;

document.getElementById("Button2").style.backgroundColor = "#a54997";

document.getElementById("Button3").disabled = false;

document.getElementById("Button3").style.backgroundColor = "#a54997";

document.getElementById("Button4").disabled = false;

document.getElementById("Button4").style.backgroundColor = "#a54997";

document.getElementById("Button5").disabled = false;

document.getElementById("Button5").style.backgroundColor = "#a54997";

for (let k=0;k<bars.length;k++) {

bars[k].style.backgroundColor="rgb(236, 190, 53)";

}

compare.textContent=' ' + " No of Comparisons: ";

terminate = !terminate;

return;

}

l5.style.backgroundColor = "cyan";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

bars[j].style.backgroundColor = "Red";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

bars[j+1].style.height = bars[j].style.height;

bars[j+1].childNodes[0].innerText = bars[j].childNodes[0].innerText;

bars[j].style.backgroundColor = "rgb(236, 190, 53)";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

l5.style.backgroundColor = null;

l6.style.backgroundColor = "cyan";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

j=j-1;

l6.style.backgroundColor = null;

compare.textContent=' ' + " No of Comparisons: "+count;

}

count++;

l4.style.backgroundColor = null;

l7.style.backgroundColor = "cyan";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

bars[j+1].style.height = h;

bars[j+1].childNodes[0].innerText = key;

bars[i].style.backgroundColor = "rgb(236, 190, 53)";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

l7.style.backgroundColor = null;

l2.style.backgroundColor = null;

}

l1.style.backgroundColor = null;

for (let i = 0; i < bars.length; i++)

{

bars[i].style.backgroundColor = "rgb(49, 226, 13)";

}

document.getElementById("Button1").disabled = false;

document.getElementById("Button1").style.backgroundColor = "#a54997";

document.getElementById("Button2").disabled = false;

document.getElementById("Button2").style.backgroundColor = "#a54997";

document.getElementById("Button3").disabled = false;

document.getElementById("Button3").style.backgroundColor = "#a54997";

document.getElementById("Button4").disabled = false;

document.getElementById("Button4").style.backgroundColor = "#a54997";

document.getElementById("Button5").disabled = false;

document.getElementById("Button5").style.backgroundColor = "#a54997";

compare.textContent=' ' + " No of Comparisons: "+count;

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

l0.style.backgroundColor = null;

}

function delaySet() {

delay = 5000;

var s = document.getElementById("speeder");

var d = parseInt(s.value);

delay=delay/d;

}

pausePlayBtn.addEventListener('click', () => {

if (isPlaying) {

isPlaying = false;

pausePlayBtn.textContent = 'Pause';

} else {

isPlaying = true;

pausePlayBtn.textContent = 'Resume';

}

});

**SELECTION SORT -**

const container = document.querySelector(".vrsplit1");

let isPlaying = false;

const pausePlayBtn = document.getElementById("pauseButton");

let terminate = false;

var count = 0;

const compare = document.getElementById("comp");

function generatebars(num) {

container.innerHTML=""

for (let i = 0; i < num; i += 1) {

const value = Math.floor(Math.random() \* (180-9)+9) + 1;

const bar = document.createElement("div");

bar.classList.add("bar");

bar.style.height = `${value/2}%`;

const barLabel = document.createElement("label");

barLabel.classList.add("bar\_\_id");

barLabel.innerHTML = value;

if (num>80) {

barLabel.style.display='none';

}

if (num<=40) {

if (num<=10) {

barLabel.style.fontSize = 'xxx-large';

}

else if (num<=20) {

barLabel.style.fontSize = 'xx-large';

}

if (num>20 && num<=30) {

barLabel.style.fontSize = 'x-large';

}

else if (num<=40) {

barLabel.style.fontSize = 'large';

}

}

bar.appendChild(barLabel);

container.appendChild(bar);

}

}

generatebars(25);

function generate() {

var n = document.getElementById("nele");

var numele = parseInt(n.value);

if (numele>400) {

window.alert("Upper bound is 400 bars. Kindly choose a value in that range!");

n.value=400;

generate();

}

else {

generatebars(numele);

}

}

function generate2() {

const value = Math.floor(Math.random() \* 80) + 1;

generatebars(value);

}

function generatebarsWorst(num) {

container.innerHTML=""

let values = Array.from({length: num}, (\_, i) => Math.max(num - i + 9, 10));

for (let i = 0; i < num; i += 1) {

const value = values[i];

const bar = document.createElement("div");

bar.classList.add("bar");

bar.style.height = `${value/2}%`;

const barLabel = document.createElement("label");

barLabel.classList.add("bar\_\_id");

barLabel.innerHTML = value;

if (num>80) {

barLabel.style.display='none';

}

if (num<=40) {

if (num<=10) {

barLabel.style.fontSize = 'xxx-large';

}

else if (num<=20) {

barLabel.style.fontSize = 'xx-large';

}

if (num>20 && num<=30) {

barLabel.style.fontSize = 'x-large';

}

else if (num<=40) {

barLabel.style.fontSize = 'large';

}

}

bar.appendChild(barLabel);

container.appendChild(bar);

}

}

function generatebarsBest(num) {

container.innerHTML=""

let values = Array.from({length: num}, (\_, i) => i + 1 + 9);

for (let i = 0; i < num; i += 1) {

const value = values[i];

const bar = document.createElement("div");

bar.classList.add("bar");

bar.style.height = `${value/2}%`;

const barLabel = document.createElement("label");

barLabel.classList.add("bar\_\_id");

barLabel.innerHTML = value;

if (num>80) {

barLabel.style.display='none';

}

if (num<=40) {

if (num<=10) {

barLabel.style.fontSize = 'xxx-large';

}

else if (num<=20) {

barLabel.style.fontSize = 'xx-large';

}

if (num>20 && num<=30) {

barLabel.style.fontSize = 'x-large';

}

else if (num<=40) {

barLabel.style.fontSize = 'large';

}

}

bar.appendChild(barLabel);

container.appendChild(bar);

}

}

function generatebest() {

const value = Math.floor(Math.random() \* 80) + 1;

generatebarsBest(value);

}

function generateworst() {

const value = Math.floor(Math.random() \* 80) + 1;

generatebarsWorst(value);

}

function disable() {

document.getElementById("Button1").disabled = true;

document.getElementById("Button1").style.backgroundColor = "#d8b6ff";

document.getElementById("Button2").disabled = true;

document.getElementById("Button2").style.backgroundColor = "#d8b6ff";

document.getElementById("Button3").disabled = true;

document.getElementById("Button3").style.backgroundColor = "#d8b6ff";

document.getElementById("Button4").disabled = true;

document.getElementById("Button4").style.backgroundColor = "#d8b6ff";

document.getElementById("Button5").disabled = true;

document.getElementById("Button5").style.backgroundColor = "#d8b6ff";

}

var delay = 5000;

async function selectionSort() {

count = 0;

let bars = document.querySelectorAll(".bar");

var l0 = document.getElementById("line0");

var l1 = document.getElementById("line1");

var l2 = document.getElementById("line2");

var l3 = document.getElementById("line3");

var l4 = document.getElementById("line4");

var l5 = document.getElementById("line5");

var l6 = document.getElementById("line6");

var l7 = document.getElementById("line7");

l0.style.backgroundColor = "lightgreen";

for (var i = 0; i < bars.length - 1; i++) {

l1.style.backgroundColor = "cyan";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

l1.style.backgroundColor = null;

var min\_idx = i;

bars[i].style.backgroundColor="red";

l2.style.backgroundColor = "cyan";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

l2.style.backgroundColor = null;

for (var j = i + 1; j < bars.length; j++) {

l3.style.backgroundColor = "cyan";

while (isPlaying) {

if (terminate) {

l3.style.backgroundColor = null;

l0.style.backgroundColor = null;

document.getElementById("Button1").disabled = false;

document.getElementById("Button1").style.backgroundColor = "#a54997";

document.getElementById("Button2").disabled = false;

document.getElementById("Button2").style.backgroundColor = "#a54997";

document.getElementById("Button3").disabled = false;

document.getElementById("Button3").style.backgroundColor = "#a54997";

document.getElementById("Button4").disabled = false;

document.getElementById("Button4").style.backgroundColor = "#a54997";

document.getElementById("Button5").disabled = false;

document.getElementById("Button5").style.backgroundColor = "#a54997";

isPlaying = false;

pausePlayBtn.textContent = 'Pause';

compare.textContent=' ' + " No of Comparisons: ";

for (let k=0;k<bars.length;k++) {

bars[k].style.backgroundColor="rgb(236, 190, 53)";

}

terminate = !terminate;

return;

}

await new Promise((resolve) => setTimeout(() => { resolve(); }, 1000));

}

if (terminate) {

l3.style.backgroundColor = null;

l0.style.backgroundColor = null;

document.getElementById("Button1").disabled = false;

document.getElementById("Button1").style.backgroundColor = "#a54997";

document.getElementById("Button2").disabled = false;

document.getElementById("Button2").style.backgroundColor = "#a54997";

document.getElementById("Button3").disabled = false;

document.getElementById("Button3").style.backgroundColor = "#a54997";

document.getElementById("Button4").disabled = false;

document.getElementById("Button4").style.backgroundColor = "#a54997";

document.getElementById("Button5").disabled = false;

document.getElementById("Button5").style.backgroundColor = "#a54997";

for (let k=0;k<bars.length;k++) {

bars[k].style.backgroundColor="rgb(236, 190, 53)";

}

compare.textContent=' ' + " No of Comparisons: ";

terminate = !terminate;

return;

}

var value1 = parseInt(bars[j].childNodes[0].innerHTML);

var value2 = parseInt(bars[min\_idx].childNodes[0].innerHTML);

if (value1 < value2) {

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

l4.style.backgroundColor = "cyan";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

l4.style.backgroundColor = null;

if(min\_idx!=i){

bars[min\_idx].style.backgroundColor="rgb(236, 190, 53)";

}

min\_idx = j;

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

bars[j].style.backgroundColor="red";

l5.style.backgroundColor = "cyan";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

l5.style.backgroundColor = null;

}

count++;

}

l6.style.backgroundColor = "cyan";

l3.style.backgroundColor = null;

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

l6.style.backgroundColor = null;

if (min\_idx != i) {

bars[min\_idx].style.backgroundColor = "red";

bars[i].style.backgroundColor = "red";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

var temp1 = bars[min\_idx].style.height;

var temp2 = bars[min\_idx].childNodes[0].innerText;

l7.style.backgroundColor = "cyan";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

bars[min\_idx].style.height = bars[i].style.height;

bars[min\_idx].childNodes[0].innerText = bars[i].childNodes[0].innerText;

bars[i].style.height = temp1;

bars[i].childNodes[0].innerText = temp2;

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

l7.style.backgroundColor = null;

bars[min\_idx].style.backgroundColor = "rgb(236, 190, 53)";

bars[i].style.backgroundColor = "rgb(236, 190, 53)";

}

bars[i].style.backgroundColor = "rgb(236, 190, 53)";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

compare.textContent=' ' + " No of Comparisons: "+count;

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

}

for (var x = 0; x < bars.length; x++) {

bars[x].style.backgroundColor = "rgb(49, 226, 13)";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

}

document.getElementById("Button1").disabled = false;

document.getElementById("Button1").style.backgroundColor = "#a54997";

document.getElementById("Button2").disabled = false;

document.getElementById("Button2").style.backgroundColor = "#a54997";

document.getElementById("Button3").disabled = false;

document.getElementById("Button3").style.backgroundColor = "#a54997";

document.getElementById("Button4").disabled = false;

document.getElementById("Button4").style.backgroundColor = "#a54997";

document.getElementById("Button5").disabled = false;

document.getElementById("Button5").style.backgroundColor = "#a54997";

compare.textContent=' ' + " No of Comparisons: "+count;

l0.style.backgroundColor = null;

}

function delaySet() {

delay = 5000;

var s = document.getElementById("speeder");

var d = parseInt(s.value);

delay=delay/d;

}

pausePlayBtn.addEventListener('click', () => {

if (isPlaying) {

isPlaying = false;

pausePlayBtn.textContent = 'Pause';

} else {

isPlaying = true;

pausePlayBtn.textContent = 'Resume';

}

});

**COUNTING SORT -**

const container = document.querySelector(".mainarray");

let C\_container = document.querySelector(".countarray");

let isPlaying = false;

const pausePlayBtn = document.getElementById("pauseButton");

let terminate = false;

var max = 0;

function generatebars(num) {

container.innerHTML="";

max = 0;

for (let i = 0; i < num; i += 1) {

const value = Math.floor(Math.random() \* (50-9)+9) + 1;

if (value > max) {

max = value;

}

const bar = document.createElement("div");

bar.classList.add("bar");

bar.style.height = `${1.5\*value}%`;

const barLabel = document.createElement("label");

barLabel.classList.add("bar\_\_id");

barLabel.innerHTML = value;

if (num<=40) {

if (num<=10) {

barLabel.style.fontSize = 'xxx-large';

}

else if (num<=20) {

barLabel.style.fontSize = 'xx-large';

}

if (num>20 && num<=30) {

barLabel.style.fontSize = 'x-large';

}

else if (num<=40) {

barLabel.style.fontSize = 'large';

}

}

bar.appendChild(barLabel);

container.appendChild(bar);

}

}

generatebars(25);

function generate() {

var n = document.getElementById("nele");

var numele = parseInt(n.value);

if (numele>100) {

window.alert("Upper bound is 100 bars. Kindly choose a value in that range!");

n.value=100;

generate();

}

else {

generatebars(numele);

}

}

function generate2() {

const value = Math.floor(Math.random() \* 80) + 1;

generatebars(value);

}

function generatebarsBest(num) {

container.innerHTML="";

const a = Math.floor(Math.random() \* (50-9)+9) + 1;

let values = Array(num).fill().map(() => a);

for (let i = 0; i < num; i += 1) {

const value = values[i];

const bar = document.createElement("div");

bar.classList.add("bar");

bar.style.height = `${1.5\*value}%`;

const barLabel = document.createElement("label");

barLabel.classList.add("bar\_\_id");

barLabel.innerHTML = value;

if (num>80) {

barLabel.style.display='none';

}

if (num<=40) {

if (num<=10) {

barLabel.style.fontSize = 'xxx-large';

}

else if (num<=20) {

barLabel.style.fontSize = 'xx-large';

}

if (num>20 && num<=30) {

barLabel.style.fontSize = 'x-large';

}

else if (num<=40) {

barLabel.style.fontSize = 'large';

}

}

bar.appendChild(barLabel);

container.appendChild(bar);

}

}

function generatebest() {

const value = Math.floor(Math.random() \* 70) + 1;

generatebarsBest(value);

}

function generateCountArray() {

C\_container.innerHTML="";

for (let i = 10; i <= max; i += 1) {

const bar = document.createElement("div");

bar.classList.add("bar2");

bar.style.height = `${1.5\*i}%`;

const barLabel = document.createElement("label");

barLabel.classList.add("bar\_\_id");

barLabel.innerHTML = i+","+00;

bar.appendChild(barLabel);

C\_container.appendChild(bar);

}

}

function disable() {

document.getElementById("Button1").disabled = true;

document.getElementById("Button1").style.backgroundColor = "#d8b6ff";

document.getElementById("Button2").disabled = true;

document.getElementById("Button2").style.backgroundColor = "#d8b6ff";

document.getElementById("Button3").disabled = true;

document.getElementById("Button3").style.backgroundColor = "#d8b6ff";

document.getElementById("Button4").disabled = true;

document.getElementById("Button4").style.backgroundColor = "#d8b6ff";

document.getElementById("Button5").disabled = true;

document.getElementById("Button5").style.backgroundColor = "#d8b6ff";

}

var delay = 5000;

async function CountingSort() {

var l0 = document.getElementById("line0");

var l1 = document.getElementById("line1");

var l2 = document.getElementById("line2");

var l3 = document.getElementById("line3");

var l4 = document.getElementById("line4");

var l5 = document.getElementById("line5");

var l6 = document.getElementById("line6");

var l7 = document.getElementById("line7");

var l8 = document.getElementById("line8");

l0.style.backgroundColor="lightgreen";

l1.style.backgroundColor="lightgreen";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

generateCountArray();

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

l1.style.backgroundColor=null;

let bars = document.querySelectorAll(".bar");

let countA = document.querySelectorAll(".bar2");

var countB = Array.from({length: 51}, (\_, i) => 0);

while (isPlaying) {

if (terminate) {

l0.style.backgroundColor=null;

document.getElementById("Button1").disabled = false;

document.getElementById("Button1").style.backgroundColor = "#a54997";

document.getElementById("Button2").disabled = false;

document.getElementById("Button2").style.backgroundColor = "#a54997";

document.getElementById("Button3").disabled = false;

document.getElementById("Button3").style.backgroundColor = "#a54997";

document.getElementById("Button4").disabled = false;

document.getElementById("Button4").style.backgroundColor = "#a54997";

document.getElementById("Button5").disabled = false;

document.getElementById("Button5").style.backgroundColor = "#a54997";

isPlaying = false;

pausePlayBtn.textContent = 'Pause';

terminate = !terminate;

compare.textContent=' ' + " No of Comparisons: ";

for (let k=0;k<bars.length;k++) {

bars[k].style.backgroundColor="rgb(236, 190, 53)";

}

terminate = !terminate;

return;

}

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

}

if (terminate) {

l0.style.backgroundColor=null;

document.getElementById("Button1").disabled = false;

document.getElementById("Button1").style.backgroundColor = "#a54997";

document.getElementById("Button2").disabled = false;

document.getElementById("Button2").style.backgroundColor = "#a54997";

document.getElementById("Button3").disabled = false;

document.getElementById("Button3").style.backgroundColor = "#a54997";

document.getElementById("Button4").disabled = false;

document.getElementById("Button4").style.backgroundColor = "#a54997";

document.getElementById("Button5").disabled = false;

document.getElementById("Button5").style.backgroundColor = "#a54997";

for (let k=0;k<bars.length;k++) {

bars[k].style.backgroundColor="rgb(236, 190, 53)";

}

compare.textContent=' ' + " No of Comparisons: ";

terminate = !terminate;

return;

}

l2.style.backgroundColor="lightgreen";

for (var i=0;i<bars.length;i++) {

++countB[bars[i].childNodes[0].innerHTML];

bars[i].style.backgroundColor="red";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

countA[bars[i].childNodes[0].innerHTML - 10].style.backgroundColor="red";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

countA[bars[i].childNodes[0].innerHTML - 10].childNodes[0].innerHTML = (bars[i].childNodes[0].innerHTML)+","+countB[bars[i].childNodes[0].innerHTML];

l3.style.backgroundColor="lightgreen";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

countA[bars[i].childNodes[0].innerHTML - 10].style.backgroundColor="rgb(236, 190, 53)";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

l3.style.backgroundColor=null;

bars[i].style.backgroundColor="rgb(236, 190, 53)";

while (isPlaying) {

if (terminate) {

l0.style.backgroundColor=null;

l2.style.backgroundColor=null;

document.getElementById("Button1").disabled = false;

document.getElementById("Button1").style.backgroundColor = "#a54997";

document.getElementById("Button2").disabled = false;

document.getElementById("Button2").style.backgroundColor = "#a54997";

document.getElementById("Button3").disabled = false;

document.getElementById("Button3").style.backgroundColor = "#a54997";

document.getElementById("Button4").disabled = false;

document.getElementById("Button4").style.backgroundColor = "#a54997";

document.getElementById("Button5").disabled = false;

document.getElementById("Button5").style.backgroundColor = "#a54997";

isPlaying = false;

pausePlayBtn.textContent = 'Pause';

terminate = !terminate;

compare.textContent=' ' + " No of Comparisons: ";

for (let k=0;k<bars.length;k++) {

bars[k].style.backgroundColor="rgb(236, 190, 53)";

}

terminate = !terminate;

return;

}

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

}

if (terminate) {

l0.style.backgroundColor=null;

l2.style.backgroundColor=null;

document.getElementById("Button1").disabled = false;

document.getElementById("Button1").style.backgroundColor = "#a54997";

document.getElementById("Button2").disabled = false;

document.getElementById("Button2").style.backgroundColor = "#a54997";

document.getElementById("Button3").disabled = false;

document.getElementById("Button3").style.backgroundColor = "#a54997";

document.getElementById("Button4").disabled = false;

document.getElementById("Button4").style.backgroundColor = "#a54997";

document.getElementById("Button5").disabled = false;

document.getElementById("Button5").style.backgroundColor = "#a54997";

for (let k=0;k<bars.length;k++) {

bars[k].style.backgroundColor="rgb(236, 190, 53)";

}

compare.textContent=' ' + " No of Comparisons: ";

terminate = !terminate;

return;

}

}

l2.style.backgroundColor=null;

for (var i=11;i<countB.length;i++) {

countB[i] += countB[i-1];;

}

l4.style.backgroundColor="lightgreen";

for (var i=0;i<countA.length;i++) {

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

countA[i].style.backgroundColor="red";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

countA[i].childNodes[0].innerHTML = (i+10)+","+countB[i+10];

l5.style.backgroundColor="lightgreen";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

countA[i].style.backgroundColor="rgb(236, 190, 53)";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

l5.style.backgroundColor=null;

while (isPlaying) {

if (terminate) {

l0.style.backgroundColor=null;

l4.style.backgroundColor=null;

document.getElementById("Button1").disabled = false;

document.getElementById("Button1").style.backgroundColor = "#a54997";

document.getElementById("Button2").disabled = false;

document.getElementById("Button2").style.backgroundColor = "#a54997";

document.getElementById("Button3").disabled = false;

document.getElementById("Button3").style.backgroundColor = "#a54997";

document.getElementById("Button4").disabled = false;

document.getElementById("Button4").style.backgroundColor = "#a54997";

document.getElementById("Button5").disabled = false;

document.getElementById("Button5").style.backgroundColor = "#a54997";

isPlaying = false;

pausePlayBtn.textContent = 'Pause';

terminate = !terminate;

compare.textContent=' ' + " No of Comparisons: ";

for (let k=0;k<bars.length;k++) {

bars[k].style.backgroundColor="rgb(236, 190, 53)";

}

terminate = !terminate;

return;

}

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

}

if (terminate) {

l0.style.backgroundColor=null;

l4.style.backgroundColor=null;

document.getElementById("Button1").disabled = false;

document.getElementById("Button1").style.backgroundColor = "#a54997";

document.getElementById("Button2").disabled = false;

document.getElementById("Button2").style.backgroundColor = "#a54997";

document.getElementById("Button3").disabled = false;

document.getElementById("Button3").style.backgroundColor = "#a54997";

document.getElementById("Button4").disabled = false;

document.getElementById("Button4").style.backgroundColor = "#a54997";

document.getElementById("Button5").disabled = false;

document.getElementById("Button5").style.backgroundColor = "#a54997";

for (let k=0;k<bars.length;k++) {

bars[k].style.backgroundColor="rgb(236, 190, 53)";

}

compare.textContent=' ' + " No of Comparisons: ";

terminate = !terminate;

return;

}

}

l4.style.backgroundColor=null;

let ans = [];

let countB\_copy = countB.slice();

for (var i=bars.length-1;i>=0;i--) {

ans[countB\_copy[bars[i].childNodes[0].innerHTML]-1] = bars[i].childNodes[0].innerHTML;

countB\_copy[i]--;

}

let array = [];

for (var i = 0;i<bars.length;i++) {

array[i]=bars[i].childNodes[0].innerHTML;

}

l6.style.backgroundColor="lightgreen";

for (var i = bars.length - 1;i>=0;i--) {

countA[array[i] - 10].style.backgroundColor="red";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

bars[countB[array[i]]-1].style.backgroundColor="red";

l7.style.backgroundColor="lightgreen";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

bars[countB[array[i]]-1].style.height=`${1.5\*array[i]}%`;

bars[countB[array[i]]-1].childNodes[0].innerHTML=array[i];

l7.style.backgroundColor=null;

var t = countB[array[i]];

countB[array[i]]--;

l8.style.backgroundColor="lightgreen";

countA[array[i]-10].childNodes[0].innerHTML = array[i]+","+countB[array[i]-1];

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

l8.style.backgroundColor=null;

countA[array[i]-10].style.backgroundColor="rgb(236, 190, 53)";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

bars[t-1].style.backgroundColor="rgb(236, 190, 53)";

while (isPlaying) {

if (terminate) {

l0.style.backgroundColor=null;

l6.style.backgroundColor=null;

document.getElementById("Button1").disabled = false;

document.getElementById("Button1").style.backgroundColor = "#a54997";

document.getElementById("Button2").disabled = false;

document.getElementById("Button2").style.backgroundColor = "#a54997";

document.getElementById("Button3").disabled = false;

document.getElementById("Button3").style.backgroundColor = "#a54997";

document.getElementById("Button4").disabled = false;

document.getElementById("Button4").style.backgroundColor = "#a54997";

document.getElementById("Button5").disabled = false;

document.getElementById("Button5").style.backgroundColor = "#a54997";

isPlaying = false;

pausePlayBtn.textContent = 'Pause';

terminate = !terminate;

compare.textContent=' ' + " No of Comparisons: ";

for (let k=0;k<bars.length;k++) {

bars[k].style.backgroundColor="rgb(236, 190, 53)";

}

terminate = !terminate;

return;

}

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

}

if (terminate) {

l0.style.backgroundColor=null;

l6.style.backgroundColor=null;

document.getElementById("Button1").disabled = false;

document.getElementById("Button1").style.backgroundColor = "#a54997";

document.getElementById("Button2").disabled = false;

document.getElementById("Button2").style.backgroundColor = "#a54997";

document.getElementById("Button3").disabled = false;

document.getElementById("Button3").style.backgroundColor = "#a54997";

document.getElementById("Button4").disabled = false;

document.getElementById("Button4").style.backgroundColor = "#a54997";

document.getElementById("Button5").disabled = false;

document.getElementById("Button5").style.backgroundColor = "#a54997";

for (let k=0;k<bars.length;k++) {

bars[k].style.backgroundColor="rgb(236, 190, 53)";

}

compare.textContent=' ' + " No of Comparisons: ";

terminate = !terminate;

return;

}

}

l6.style.backgroundColor=null;

for (let i = 0; i < bars.length; i++) {

bars[i].style.backgroundColor = "rgb(49, 226, 13)";

await new Promise((resolve) => setTimeout(() => { resolve(); }, delay));

}

document.getElementById("Button1").disabled = false;

document.getElementById("Button1").style.backgroundColor = "#a54997";

document.getElementById("Button2").disabled = false;

document.getElementById("Button2").style.backgroundColor = "#a54997";

document.getElementById("Button3").disabled = false;

document.getElementById("Button3").style.backgroundColor = "#a54997";

document.getElementById("Button4").disabled = false;

document.getElementById("Button4").style.backgroundColor = "#a54997";

document.getElementById("Button5").disabled = false;

document.getElementById("Button5").style.backgroundColor = "#a54997";

}

function delaySet() {

delay = 5000;

var s = document.getElementById("speeder");

var d = parseInt(s.value);

delay=delay/d;

}

pausePlayBtn.addEventListener('click', () => {

if (isPlaying) {

isPlaying = false;

pausePlayBtn.textContent = 'Pause';

} else {

isPlaying = true;

pausePlayBtn.textContent = 'Resume';

}

});

**CSS FILES -**

**Font.css-**

/\* cyrillic-ext \*/

@font-face {

font-family: 'Jura';

font-style: normal;

font-weight: 600;

font-display: swap;

src: url(https://fonts.gstatic.com/s/jura/v24/z7NOdRfiaC4Vd8hhoPzfb5vBTP2D6purR\_ibHw.woff2) format('woff2');

unicode-range: U+0460-052F, U+1C80-1C88, U+20B4, U+2DE0-2DFF, U+A640-A69F, U+FE2E-FE2F;

}

/\* cyrillic \*/

@font-face {

font-family: 'Jura';

font-style: normal;

font-weight: 600;

font-display: swap;

src: url(https://fonts.gstatic.com/s/jura/v24/z7NOdRfiaC4Vd8hhoPzfb5vBTP2D6puiR\_ibHw.woff2) format('woff2');

unicode-range: U+0301, U+0400-045F, U+0490-0491, U+04B0-04B1, U+2116;

}

/\* greek-ext \*/

@font-face {

font-family: 'Jura';

font-style: normal;

font-weight: 600;

font-display: swap;

src: url(https://fonts.gstatic.com/s/jura/v24/z7NOdRfiaC4Vd8hhoPzfb5vBTP2D6puqR\_ibHw.woff2) format('woff2');

unicode-range: U+1F00-1FFF;

}

/\* greek \*/

@font-face {

font-family: 'Jura';

font-style: normal;

font-weight: 600;

font-display: swap;

src: url(https://fonts.gstatic.com/s/jura/v24/z7NOdRfiaC4Vd8hhoPzfb5vBTP2D6pulR\_ibHw.woff2) format('woff2');

unicode-range: U+0370-03FF;

}

/\* kayah-li \*/

@font-face {

font-family: 'Jura';

font-style: normal;

font-weight: 600;

font-display: swap;

src: url(https://fonts.gstatic.com/s/jura/v24/z7NOdRfiaC4Vd8hhoPzfb5vBTP2D6pvuR\_ibHw.woff2) format('woff2');

unicode-range: U+200C-200D, U+2010, U+25CC, U+A900-A92F;

}

/\* vietnamese \*/

@font-face {

font-family: 'Jura';

font-style: normal;

font-weight: 600;

font-display: swap;

src: url(https://fonts.gstatic.com/s/jura/v24/z7NOdRfiaC4Vd8hhoPzfb5vBTP2D6pupR\_ibHw.woff2) format('woff2');

unicode-range: U+0102-0103, U+0110-0111, U+0128-0129, U+0168-0169, U+01A0-01A1, U+01AF-01B0, U+1EA0-1EF9, U+20AB;

}

/\* latin-ext \*/

@font-face {

font-family: 'Jura';

font-style: normal;

font-weight: 600;

font-display: swap;

src: url(https://fonts.gstatic.com/s/jura/v24/z7NOdRfiaC4Vd8hhoPzfb5vBTP2D6puoR\_ibHw.woff2) format('woff2');

unicode-range: U+0100-024F, U+0259, U+1E00-1EFF, U+2020, U+20A0-20AB, U+20AD-20CF, U+2113, U+2C60-2C7F, U+A720-A7FF;

}

/\* latin \*/

@font-face {

font-family: 'Jura';

font-style: normal;

font-weight: 600;

font-display: swap;

src: url(https://fonts.gstatic.com/s/jura/v24/z7NOdRfiaC4Vd8hhoPzfb5vBTP2D6pumR\_g.woff2) format('woff2');

unicode-range: U+0000-00FF, U+0131, U+0152-0153, U+02BB-02BC, U+02C6, U+02DA, U+02DC, U+2000-206F, U+2074, U+20AC, U+2122, U+2191, U+2193, U+2212, U+2215, U+FEFF, U+FFFD;

}

/\* cyrillic-ext \*/

@font-face {

font-family: 'Ubuntu';

font-style: normal;

font-weight: 400;

font-display: swap;

src: url(https://fonts.gstatic.com/s/ubuntu/v20/4iCs6KVjbNBYlgoKcg72j00.woff2) format('woff2');

unicode-range: U+0460-052F, U+1C80-1C88, U+20B4, U+2DE0-2DFF, U+A640-A69F, U+FE2E-FE2F;

}

/\* cyrillic \*/

@font-face {

font-family: 'Ubuntu';

font-style: normal;

font-weight: 400;

font-display: swap;

src: url(https://fonts.gstatic.com/s/ubuntu/v20/4iCs6KVjbNBYlgoKew72j00.woff2) format('woff2');

unicode-range: U+0301, U+0400-045F, U+0490-0491, U+04B0-04B1, U+2116;

}

/\* greek-ext \*/

@font-face {

font-family: 'Ubuntu';

font-style: normal;

font-weight: 400;

font-display: swap;

src: url(https://fonts.gstatic.com/s/ubuntu/v20/4iCs6KVjbNBYlgoKcw72j00.woff2) format('woff2');

unicode-range: U+1F00-1FFF;

}

/\* greek \*/

@font-face {

font-family: 'Ubuntu';

font-style: normal;

font-weight: 400;

font-display: swap;

src: url(https://fonts.gstatic.com/s/ubuntu/v20/4iCs6KVjbNBYlgoKfA72j00.woff2) format('woff2');

unicode-range: U+0370-03FF;

}

/\* latin-ext \*/

@font-face {

font-family: 'Ubuntu';

font-style: normal;

font-weight: 400;

font-display: swap;

src: url(https://fonts.gstatic.com/s/ubuntu/v20/4iCs6KVjbNBYlgoKcQ72j00.woff2) format('woff2');

unicode-range: U+0100-024F, U+0259, U+1E00-1EFF, U+2020, U+20A0-20AB, U+20AD-20CF, U+2113, U+2C60-2C7F, U+A720-A7FF;

}

/\* latin \*/

@font-face {

font-family: 'Ubuntu';

font-style: normal;

font-weight: 400;

font-display: swap;

src: url(https://fonts.gstatic.com/s/ubuntu/v20/4iCs6KVjbNBYlgoKfw72.woff2) format('woff2');

unicode-range: U+0000-00FF, U+0131, U+0152-0153, U+02BB-02BC, U+02C6, U+02DA, U+02DC, U+2000-206F, U+2074, U+20AC, U+2122, U+2191, U+2193, U+2212, U+2215, U+FEFF, U+FFFD;

}

/\* latin-ext \*/

@font-face {

font-family: 'Oxygen';

font-style: normal;

font-weight: 300;

font-display: swap;

src: url(https://fonts.gstatic.com/s/oxygen/v15/2sDcZG1Wl4LcnbuCJW8zZmW5O7w.woff2) format('woff2');

unicode-range: U+0100-024F, U+0259, U+1E00-1EFF, U+2020, U+20A0-20AB, U+20AD-20CF, U+2113, U+2C60-2C7F, U+A720-A7FF;

}

/\* latin \*/

@font-face {

font-family: 'Oxygen';

font-style: normal;

font-weight: 300;

font-display: swap;

src: url(https://fonts.gstatic.com/s/oxygen/v15/2sDcZG1Wl4LcnbuCJW8zaGW5.woff2) format('woff2');

unicode-range: U+0000-00FF, U+0131, U+0152-0153, U+02BB-02BC, U+02C6, U+02DA, U+02DC, U+2000-206F, U+2074, U+20AC, U+2122, U+2191, U+2193, U+2212, U+2215, U+FEFF, U+FFFD;

}

/\* latin \*/

@font-face {

font-family: 'Electrolize';

font-style: normal;

font-weight: 400;

font-display: swap;

src: url(https://fonts.gstatic.com/s/electrolize/v14/cIf5Ma1dtE0zSiGSiED7AXEBuI8.woff2) format('woff2');

unicode-range: U+0000-00FF, U+0131, U+0152-0153, U+02BB-02BC, U+02C6, U+02DA, U+02DC, U+2000-206F, U+2074, U+20AC, U+2122, U+2191, U+2193, U+2212, U+2215, U+FEFF, U+FFFD;

}

**SideNavIcon -**

/\* fallback \*/

@font-face {

font-family: 'Material Icons';

font-style: normal;

font-weight: 400;

src: url(https://fonts.gstatic.com/s/materialicons/v139/flUhRq6tzZclQEJ-Vdg-IuiaDsNc.woff2) format('woff2');

}

.material-icons {

font-family: 'Material Icons';

font-weight: normal;

font-style: normal;

font-size: 24px;

line-height: 1;

letter-spacing: normal;

text-transform: none;

display: inline-block;

white-space: nowrap;

word-wrap: normal;

direction: ltr;

-webkit-font-feature-settings: 'liga';

-webkit-font-smoothing: antialiased;

}

**Style.css-**

:root {

--color-1: #186cb8;

--color-2: #2a9a9f;

--color-3: #f1b211;

--color-4: #e83611;

--color-5: #f9002f;

}

h1 {

font-family: "Exo", sans-serif;

font-size: 7vw;

font-weight: 700;

width: -webkit-min-content;

width: -moz-min-content;

width: min-content;

margin: auto;

text-transform: uppercase;

background: linear-gradient(219deg,

var(--color-1) 19%,

transparent 19%,transparent 20%,

var(--color-2) 20%, var(--color-2) 39%,

transparent 39%,transparent 40%,

var(--color-3) 40%,var(--color-3) 59% ,

transparent 59%,transparent 60%,

var(--color-4) 60%, var(--color-4) 79%,

transparent 79%, transparent 80%,

var(--color-5) 80%);

background-clip: text;

-webkit-background-clip: text;

color: transparent;

}

.container {

padding: 0rem;

text-align: center;

background: radial-gradient(circle at 1.4% 1.4% ,var(--color-1) .8%,transparent .8% ),

radial-gradient(circle at 5.5% 3%,var(--color-2) .45% ,transparent .45% ),

radial-gradient(circle at 2.5% 3.5%,var(--color-3) .5% ,transparent .5% ),

radial-gradient(circle at 4.5% 1.2%,var(--color-4) .25%,transparent .25% ),

radial-gradient(circle at 98% 98% ,var(--color-1) .8%,transparent .8% ),

radial-gradient(circle at 95% 95%,var(--color-2) .45% ,transparent .45% ),

radial-gradient(circle at 94.5% 97.5%,var(--color-3) .5% ,transparent .5% ),

radial-gradient(circle at 98.5% 95.5%,var(--color-4) .25%,transparent .25% );

}

**Style1.css-**

body {

background: url(background.jpeg);

background-position: left;

background-size: cover;

background-repeat: no-repeat;

background-attachment: fixed;

}

#header {

font-family: 'Oxygen', monospace, Courier;

color: white;

padding-right: 15px;

margin-top: 20vh;

font-size: 6vw;

letter-spacing: 5px;

position: relative;

text-align: right;

font-weight: bold;

}

@media only screen and (max-width: 850px) {

#header {

font-size: 10vw;

margin-right: 10%;

}

}

#cover {

font-family: 'Ubuntu', monospace, Courier;

position: relative;

width: 100%;

margin-top: 30vh;

padding-top: 5vh;

min-height: 130vh;

display: block;

overflow: auto;

color: white;

background: linear-gradient(0deg, rgba(20, 14, 8, 0.944) 0%, rgba(22,22,22,1) 70%);

word-wrap: normal;

}

#cover a:visited {

color: #d17f3f;

}

#cover a {

color: #d17f3f;

}

@media only screen and (min-height: 1081px) {

#cover {

min-height: 100vh;

}

}

#description-box {

margin: auto;

padding-right: 5vw;

padding-left: 5vw;

font-size: 25px;

max-width: 850px;

position: relative;

}

@media only screen and (max-width: 700px) {

#description-box {

font-size: 16px;

}

}

#description-box button {

background-color: #d17f3f;

outline: none;

border: none;

margin-top: 20px;

margin-bottom: 30px;

position: relative;

left: 50%;

-ms-transform: translateX(-50%);

transform: translateX(-50%);

width: 160px;

height: 60px;

border-radius: 35px;

font-size: 25px;

}

**Style2.css-**

html {

scroll-behavior: smooth;

overflow-x: hidden;

}

body {

margin: 0px;

font-family: 'Oxygen', cursive;

font-size: 25px;

color: black;

}

.sidenav {

display: block;

height: 100%; /\* Full-height: remove this if you want "auto" height \*/

width:0px;

border: none;

position: fixed; /\* Fixed Sidebar (stay in place on scroll) \*/

z-index: 6; /\* Stay on top \*/

top: 0; /\* Stay at the top \*/

left: 0;

background-color: #111; /\* Black \*/

overflow-x: hidden; /\* Disable horizontal scroll \*/

padding-top: 15px;

font-weight: bolder;

white-space: nowrap;

font-size: 28px;

transition: width 0.4s;

-webkit-touch-callout: none; /\* iOS Safari \*/

-webkit-user-select: none; /\* Safari \*/

-khtml-user-select: none; /\* Konqueror HTML \*/

-moz-user-select: none; /\* Old versions of Firefox \*/

-ms-user-select: none; /\* Internet Explorer/Edge \*/

user-select: none; /\* Non-prefixed version, currently

supported by Chrome, Edge, Opera and Firefox \*/

}

.sidenav::-webkit-scrollbar {

width: 10px;

}

.sidenav::-webkit-scrollbar-track {

display: none;

}

.sidenav::-webkit-scrollbar-thumb {

box-shadow: inset 0 0 10px 10px rgb(71, 71, 71);

border: solid 3px transparent;

border-radius: 5px;

}

.sidenav::-webkit-scrollbar-thumb:active {

box-shadow: inset 0 0 10px 10px rgb(90, 90, 90);

border: solid 3px transparent;

border-radius: 5px;

}

.show {

width: 300px;

}

@media only screen and (min-height: 1081px) {

.show {

border-right: 2px solid #afafaf;

}

}

.sidenav-title {

text-decoration: none;

padding: 8px 8px 15px 25px;

margin-bottom: 5px;

color: #818181;

display: block;

}

.sidenav-subtitle {

text-decoration: none;

padding: 8px 8px 15px 25px;

margin-bottom: 5px;

font-size: 20px;

color: #535353;

display: block;

}

.sidenav-element {

padding: 8px 8px 15px 75px;

text-decoration: none;

font-size: 25px;

color: #818181;

display: block;

transition: 0.1s;

}

.sidenav a:hover {

color: #f1f1f1;

background-color: #818181;

}

#sep {

width: 89%;

position: relative;

left: 50%;

-ms-transform: translateX(-50%);

transform: translateX(-50%);

height: 1px;

margin:8px 0px 10px 0px;

background-color: #afafaf;

}

.sidenav-btn {

background: none;

margin: 0 !important;

padding: 12px;

border: none;

outline: none !important;

vertical-align: middle;

transition: 0.1s;

}

#audio {

margin-right: 1px !important;

float: right;

}

.ripple {

background-position: center !important;

transition: background 0.2s !important;

}

.ripple:hover {

background: #5e5e5e radial-gradient(circle, transparent 1%, #404144 1%) center/15000% !important;

}

.ripple:active {

background-size: 100% !important;

transition: background 0s !important;

}

.ripple-sort-btn {

background-position: center !important;

transition: background 0.2s !important;

}

.ripple-sort-btn:hover {

background: #aa5311 radial-gradient(circle, transparent 1%, #ca7e43 1%) center/15000% !important;

}

.ripple-sort-btn:active {

background-size: 100% !important;

transition: background 0s !important;

}

.topnav {

position: fixed;

width: 100%;

top: 0;

font-family: 'Jura', monospace, Courier;

list-style-type: none;

margin: 0;

padding: 0;

overflow: hidden;

background: rgb(22, 22, 22);

z-index: 5;

-webkit-touch-callout: none;

-webkit-user-select: none;

-khtml-user-select: none;

-moz-user-select: none;

-ms-user-select: none;

user-select: none;

}

.topnav-element {

font-size: 32px;

float: left;

margin: 10px 20px 10px 10px;

color: #afafaf;

text-decoration: none;

}

.icon {

position: relative;

padding-top: 1px;

font-size: 33px !important;

display:inline-flex !important;

vertical-align: top;

color: #d17f3f;

pointer-events: none;

}

@media only screen and (max-width: 850px) {

.icon {

font-size: 22px !important;

}

.topnav-element {

font-size: 22px !important;

}

}

.symbol {

font-family: Helvetica, sans-serif;

font-weight: normal;

unicode-bidi: isolate;

font-variant-numeric: tabular-nums;

text-transform: none;

text-indent: 0px !important;

text-align: start !important;

text-align-last: start !important;

}

.footer {

width: 100%;

font-family: 'Ubuntu', monospace, Courier;

color:#afafaf;

padding-top: 30px;

padding-bottom: 50px;

position: relative;

background-color: rgb(22, 22, 22);

font-size: 20px;

white-space: nowrap;

}

.footer-container {

position: relative;

left: 50%;

-ms-transform: translateX(-50%);

transform: translateX(-50%);

display: inline-block;

}

.footer-content {

float: left;

margin-left: 6vw;

margin-bottom: 15px;

vertical-align: top;

text-decoration: none;

}

.footer-title {

color: #d17f3f;

font-size: 25px;

}

.footer-content img {

padding:5px;

width: 32px;

height: 32px;

}

@media only screen and (max-width: 820px) {

.footer {

font-size: 17px;

}

.footer-title {

font-size: 20px;

}

.footer-content:first-child {

margin-left: 30px !important;

}

}

.credits {

position: absolute;

bottom: 10px;

white-space: initial;

width: 100%;

text-align: center;

font-size:x-large;

}

.credits a {

color: #BB4430;

}

**Visualize.css -**

.screensplit {

background-color: black;

width: 100%;

height: 100vh;

}

.hrsplit1 {

background-color: black;

width:100%;

height: 80%;

}

.hrsplit2{

background-color:black;

width: 100%;

height: 20%;

}

.vrsplit1 {

background-color: black;

width: 70%;

height: 100%;

float:left;

margin: 0 auto;

display: flex;

justify-content: center;

align-items: flex-end;

}

.vrsplit2 {

background-color: black;

width:29.8%;

height: 100%;

float:left;

color: crimson;

font-size: revert;

border-width: 2px;

border-left-style: solid;

border-color: aliceblue;

border-right-style: none;

border-top-style: none;

border-bottom-style: none;

}

.split1 {

width: 70%;

height: 100%;

float: left;

background-color: bisque;

}

.split2 {

width: 29.8%;

height: 100%;

float: left;

background-color:black;

color:aliceblue;

border-width: 2px;

border-left-style: solid;

border-color: aliceblue;

border-right-style: none;

border-top-style: none;

border-bottom-style: none;

}

table,th {

border-color: aliceblue;

border: 2px solid;

}

td {

border-color: aliceblue;

border: 2px solid;

font-size: smaller;

align-items: center;

}

.bar {

background-color: rgb(236, 190, 53);

transition: 0.3s all ease;

margin: 0 1px;

flex-grow: 1;

text-align: center;

}

.bar\_\_id {

color: crimson;

font-size: xx-small;

}

.btn1 {

padding: 10px;

font-weight: bolder;

background-color: #a54997;

border-radius: 10px;

color: white;

font-size: 12px;

border: white;

margin-top: 1vw;

margin-right: 1vw;

user-select:none;

}

.btn2 {

padding: 10px;

font-weight: bolder;

background-color: #a54997;

border-radius: 10px;

color: white;

font-size: 12px;

border: white;

user-select:none;

}

body {

background: url(background.jpeg);

background-position: left;

background-size: cover;

background-repeat: no-repeat;

background-attachment: fixed;

}

#header {

font-family: 'Oxygen', monospace, Courier;

color: white;

padding-right: 15px;

margin-top: 20vh;

font-size: 6vw;

letter-spacing: 5px;

position: relative;

text-align: right;

font-weight: bold;

}

@media only screen and (max-width: 850px) {

#header {

font-size: 10vw;

margin-right: 10%;

}

}

#cover {

font-family: 'Ubuntu', monospace, Courier;

position: relative;

width: 100%;

min-height: 130vh;

display: block;

overflow: auto;

color: white;

background: linear-gradient(0deg, rgba(20, 14, 8, 0.944) 0%, rgba(22,22,22,1) 70%);

word-wrap: normal;

}

#cover a:visited {

color:darkgoldenrod;

}

#cover a {

color: #d17f3f;

}

@media only screen and (min-height: 1081px) {

#cover {

min-height: 100vh;

}

}

#description-box {

margin: auto;

padding-right: 5vw;

padding-left: 5vw;

font-size: 25px;

max-width: 850px;

position: relative;

}

@media only screen and (max-width: 700px) {

#description-box {

font-size: 16px;

}

}

#description-box button {

background-color: #d17f3f;

outline: none;

border: none;

margin-top: 20px;

margin-bottom: 30px;

position: relative;

left: 50%;

-ms-transform: translateX(-50%);

transform: translateX(-50%);

width: 160px;

height: 60px;

border-radius: 35px;

font-size: 25px;

}

**Visualize\_linear.css-**

.screensplit {

background-color: black;

width: 100%;

height: 100vh;

}

.hrsplit1 {

background-color: black;

width:100%;

height: 80%;

}

.hrsplit2{

background-color:black;

width: 100%;

height: 20%;

}

.vrsplit1 {

background-color: black;

width: 70%;

height: 100%;

float:left;

margin: 0 auto;

}

.mainarray {

background-color: black;

width: 100%;

height: 50%;

margin: 0 auto;

display: flex;

justify-content: center;

align-items: flex-end;

}

.countarray {

background-color: black;

width: 100%;

height: 50%;

margin: 0 auto;

display: flex;

justify-content: center;

align-items: flex-end;

}

.vrsplit2 {

background-color: black;

width:29.8%;

height: 100%;

float:left;

color: crimson;

font-size: x-large;

border-width: 2px;

border-left-style: solid;

border-color: aliceblue;

border-right-style: none;

border-top-style: none;

border-bottom-style: none;

}

.split1 {

width: 70%;

height: 100%;

float: left;

background-color: bisque;

}

.split2 {

width: 29.8%;

height: 100%;

float: left;

background-color:black;

color:aliceblue;

border-width: 2px;

border-left-style: solid;

border-color: aliceblue;

border-right-style: none;

border-top-style: none;

border-bottom-style: none;

}

table,th {

border-color: aliceblue;

border: 2px solid;

}

td {

border-color: aliceblue;

border: 2px solid;

font-size: smaller;

align-items: center;

}

.bar2 {

background-color: rgb(236, 190, 53);

transition: 0.3s all ease;

margin: 0 1px;

flex-grow: 1;

text-align: center;

}

.bar {

background-color: rgb(236, 190, 53);

transition: 0.3s all ease;

margin: 0 1px;

flex-grow: 1;

text-align: center;

}

.bar\_\_id {

color: crimson;

font-size: xx-small;

}

.btn1 {

padding: 10px;

font-weight: bolder;

background-color: #a54997;

border-radius: 10px;

color: white;

font-size: 12px;

border: white;

margin-top: 1vw;

margin-right: 1vw;

user-select: none;

}

.btn2 {

padding: 10px;

font-weight: bolder;

background-color: #a54997;

border-radius: 10px;

color: white;

font-size: 12px;

border: white;

user-select: none;

}

body {

background: url(background.jpeg);

background-position: left;

background-size: cover;

background-repeat: no-repeat;

background-attachment: fixed;

}

#header {

font-family: 'Oxygen', monospace, Courier;

color: white;

padding-right: 15px;

margin-top: 20vh;

font-size: 6vw;

letter-spacing: 5px;

position: relative;

text-align: right;

font-weight: bold;

}

@media only screen and (max-width: 850px) {

#header {

font-size: 10vw;

margin-right: 10%;

}

}

#cover {

font-family: 'Ubuntu', monospace, Courier;

position: relative;

width: 100%;

min-height: 130vh;

display: block;

overflow: auto;

color: white;

background: linear-gradient(0deg, rgba(20, 14, 8, 0.944) 0%, rgba(22,22,22,1) 70%);

word-wrap: normal;

}

#cover a:visited {

color:darkgoldenrod;

}

#cover a {

color: #d17f3f;

}

@media only screen and (min-height: 1081px) {

#cover {

min-height: 100vh;

}

}

#description-box {

margin: auto;

padding-right: 5vw;

padding-left: 5vw;

font-size: 25px;

max-width: 850px;

position: relative;

}

@media only screen and (max-width: 700px) {

#description-box {

font-size: 16px;

}

}

#description-box button {

background-color: #d17f3f;

outline: none;

border: none;

margin-top: 20px;

margin-bottom: 30px;

position: relative;

left: 50%;

-ms-transform: translateX(-50%);

transform: translateX(-50%);

width: 160px;

height: 60px;

border-radius: 35px;

font-size: 25px;

}

**FEEDBACK FORM-**

**Html file -**

<!DOCTYPE html>

<html>

<head>

<title>

Feedback Form

</title>

<link rel="stylesheet" href="feedbackcss.css">

<script>

function validateForm() {

var name = document.forms["feedbackForm"]["name"].value;

var age = document.forms["feedbackForm"]["name"].value;

var email = document.forms["feedbackForm"]["email"].value;

var message = document.forms["feedbackForm"]["message"].value;

var regEmail = /^\w+([\.-]?\w+)\*@\w+([\.-]?\w+)\*(\.\w{2,3})+$/g; //Javascript reGex for Email Validation.

var regName = /\d+$/g; // Javascript reGex for Name validation

if (name == "" || regName.test(name)) {

alert("Name must be filled out");

return false;

}

if (age == "") {

alert("Name must be filled out correctly");

return false;

}

if (email == "" || !regEmail.test(email)) {

alert("Email must be filled out correctly");

return false;

}

if (message == "") {

alert("Message must be filled out");

return false;

}

return true;

}

</script>

</head>

<body background="bg.jpg" style="background-repeat:no-repeat; background-size:100% 100vh;">

<center>

<div id="div1" style="background-color: rgba(0, 0, 0, 0.7)">

<h1 font-size : 70px style="margin:20px;">Feedback Form</h1>

<form name="feedbackform" action="mailto:rishabh4087@rla.du.ac.in" method="post"

onsubmit="return validateForm()" enctype="text/plain">

<label for="name">Name:</label>

<input type="text" id="name" name="name" required><br><br>

<label for"age">Age:</label>

<input type="number" min="15" max="100" name="Age" required><br><br>

<label for="email">Email:</label>

<input type="email" id="email" name="email" required><br><br>

<label for="" text-align:left>Rating:</label>

<div id="response">

<input type="radio" id="sad" name="response" value="sad" class="radio" />

<label class="emoji emoji1" for="sad">&#128542</label>

<input type="radio" name="response" id="average" value="average" class="radio" />

<label class="emoji emoji2" for="average">&#128528</label>

<input type="radio" name="response" id="happy" value="happy" class="radio" />

<label class="emoji emoji3" for="happy"> &#128522</label>

</div>

<br>

<label id="label1" for="message">Message</label> <br>

<textarea id="message" name="message" rows="7" cols="40"></textarea><br><br>

<input type="submit" value="Submit" class="button">

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

<input type="reset" value="Reset" class="button">

</form>

</div>

</center>

</body>

</html>

**CSS FILE -**

\* {

margin: 0;

padding: 0;

font-family: sans-serif;

}

#div1 {

background-color: black;

color: white;

font-size: larger;

border-radius: 15px;

width: 70%;

position: absolute;

top: 15%;

left: 20%;

margin: -50px 0 0 -50px;

}

#div5 {

display: flex;

align-items: center;

justify-content: center;

}

.radio {

display: none;

}

.emoji {

font-size: 40px;

padding: 5px 10px;

cursor: pointer;

display: inline-block;

}

.button{

height: 40px;

width: 70px;

margin:20px;

background:burlywood;

}

.button:hover {

color: rgba(255, 0, 0, 1);

box-shadow: 0 5px 15px rgba(145, 92, 182, .4);

}

#sad:checked~.emoji1,

#average:checked~.emoji2,

#happy:checked~.emoji3 {

background-color: blanchedalmond;

}

@media (min-width: 1700px) {

#div1 {

font-size:25px;

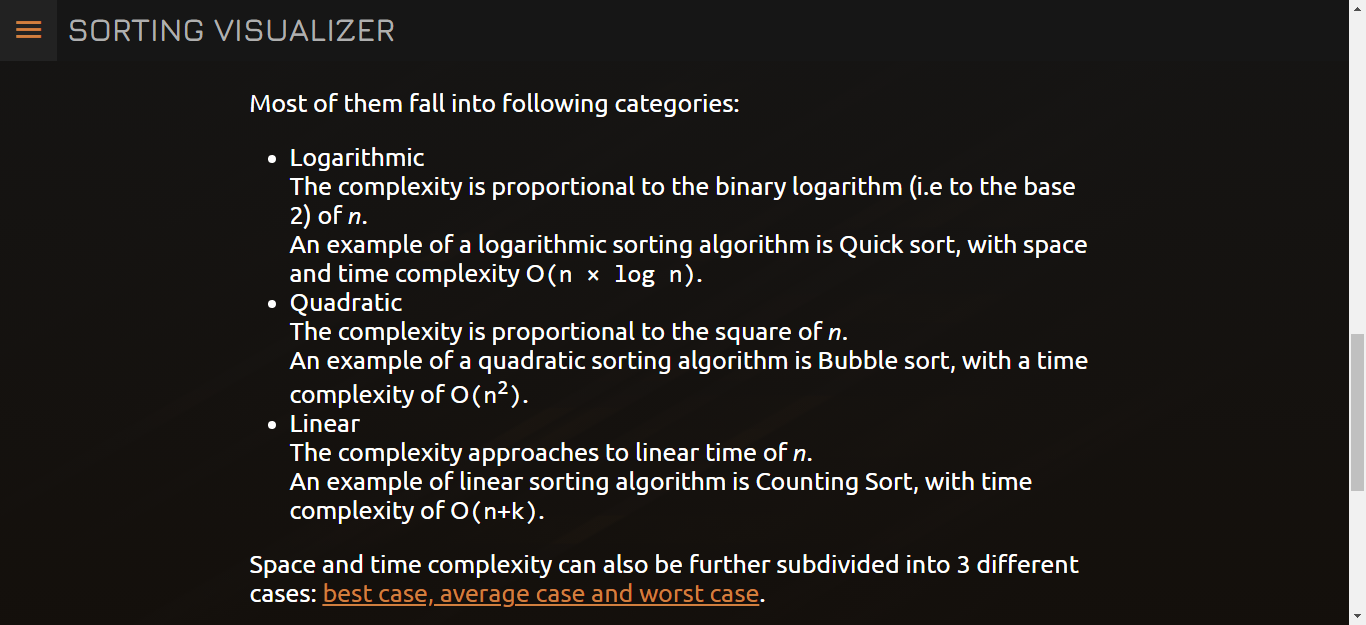
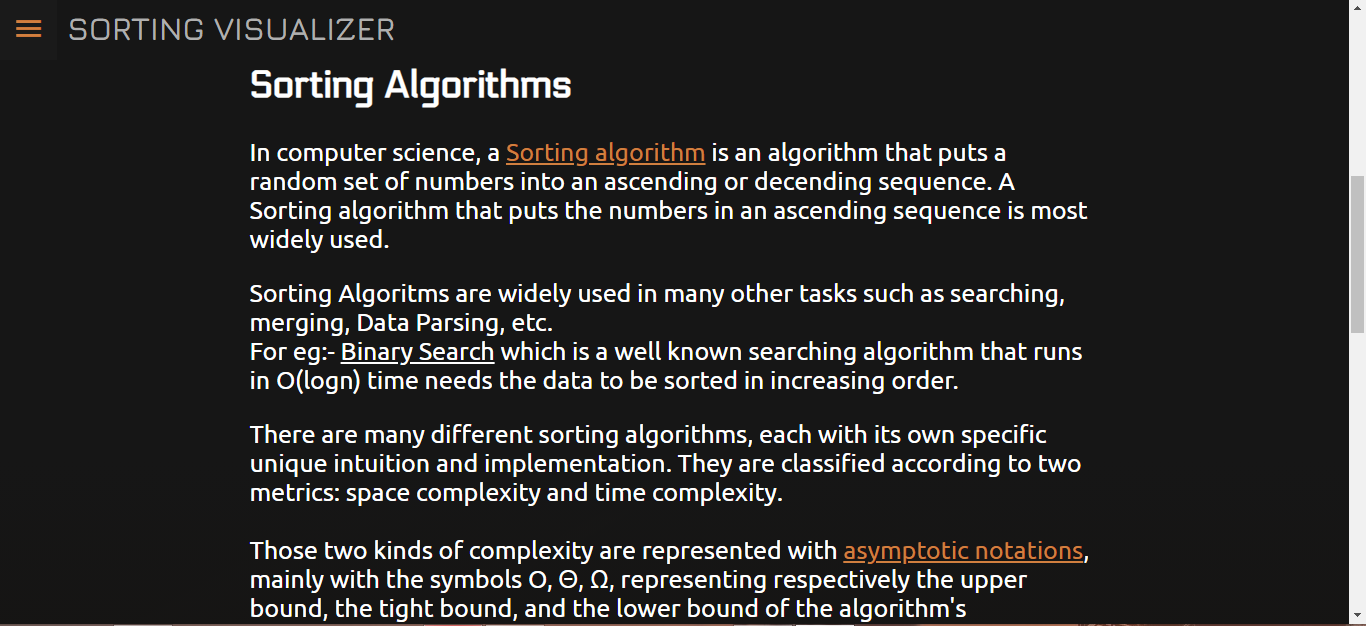
top: 20%;

}

}

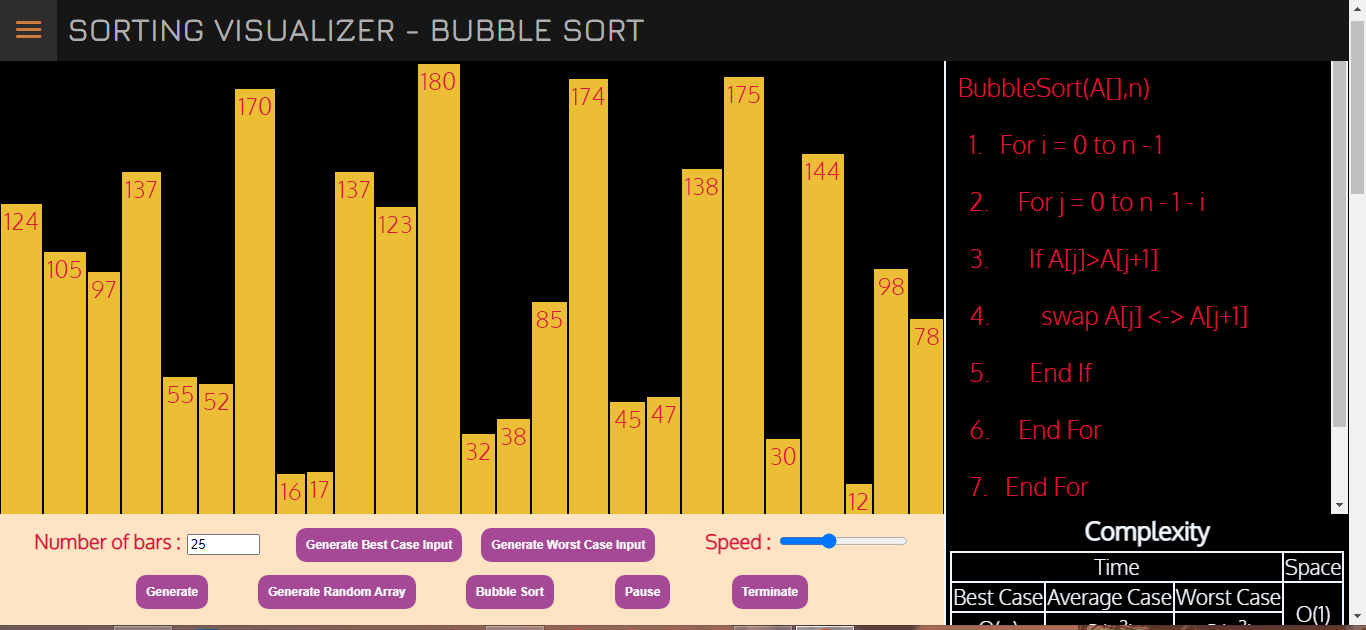
**OUTPUT-**

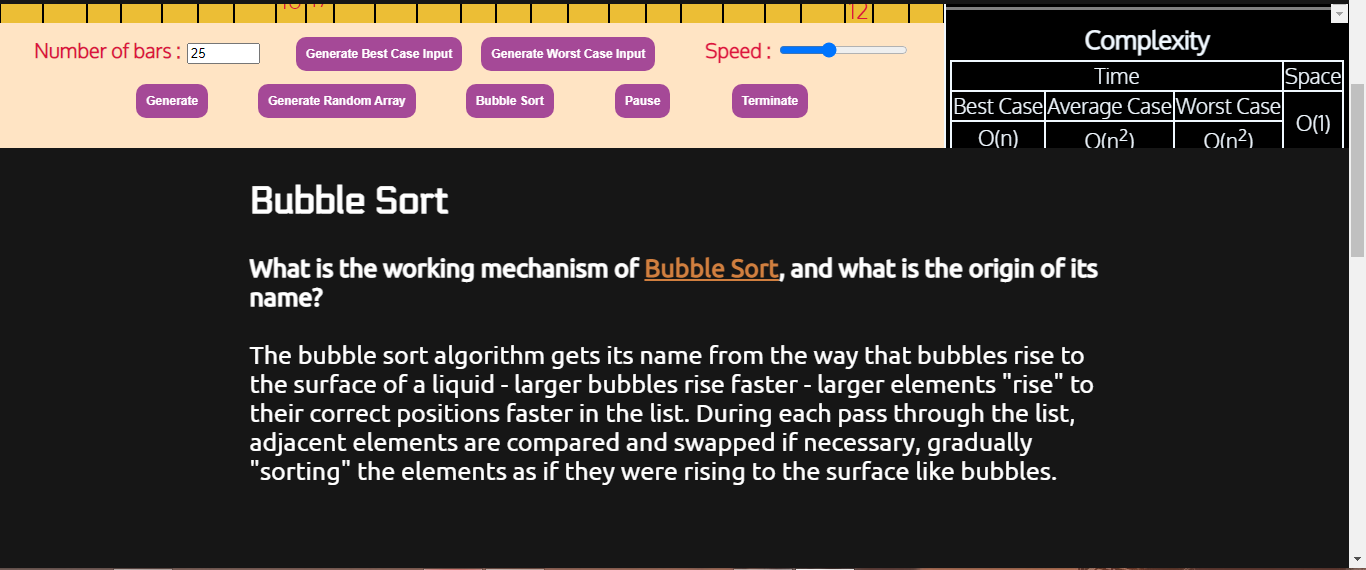








****

****

