

**Puranmal Lahoti Government Polytechnic ,Latur.**

**A**

**Micro Project On**

# “ SIMULATION OF SORTING ”

**In**

**Information Technology**

**By**

**APURVA BARDAPURKAR**

**TANUJA BUGE**

**KAWATHE MADHURA**

**KANHOPATRA KENDRE**

**Under The Guidence Of**

**Mrs. A.S.SHINDE**

**Department Of Information Technology**

**Puranmal Lahoti Government Polytechnic Latur**

**Information Technology Department**

**Subject Name: Client Side Scripting**

**Semester: 5th**

**Group No: 08**

**Title: Simulation Of Sorting.**

**Team Details:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.**  **No** | **Roll. No** | **Enrollment No** | **Name** |
| 1 | 03 | 2000160343 | BARDAPURKAR APURVA |
| 2 | 05 | 2000160345 | BUGE TANUJA |
| 3 | 22 | 2000160364 | KAWATHE MADHURA |
| 4 | 23 | 2000160365 | KENDRE KANHOPATRA |

## Part - A

Title of Micro-Project

### “ SIMULATION OF SORTING ”

**1.0 Aim of project:**

Develop webpage for Simulation of sorting.

**2.0 Course Outcomes :**

* Create interactive web pages using program flow control structure.
* Create interactive webpage using functions.
* Create event based web forms using Java script.

**3.0. Proposed Methodology :**

This Document Describes Simulation of Sorting .

1. Colored representation of step being executed. **Blue:**default . **Yellow:** Being compared **Red:** Identified as in incorrect position and to be moved .**Green:** In correct position.
2. Controls for visualizations 2.1) Speed of visualization (5 speed levels) 2.2) Data size () 2.3) Generation of new data (Randomly generate new data).
3. Time and Space complexity of algorithm being visualized.

**4.0 Action Plan:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.**  **No** | **Details of Activity** | **Planned**  **Start date** | **Planned**  **Finish date** | **Name of**  **members** |
| 1 | Selection of microproject  title | 18/10/2022 | 23/10/2022 | All  Members |
| 2 | Planning and requirement gathering of microproject | 24/10/2022 | 30/10/2022 | Kawathe  Madhura,  Buge  Tanuja |
| 3 | Devloping code | 30/10/2022 | 03/11/2022 | Kanhopatra  Kendre,  Apurva  Bardapurkar |
| 4 | Preparing document and report of project | 03/11/2022 | 05/11/2022 | All  members |

**5.0 Resources Required :**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Name of**  **Resources** | **Specifications** | **Quantity** |
| 1 | Computer | Processor-i5 RAM-8GB  HDD-1TB | 1 |
| 2 | Software | Visual studio,Notepad, Microsoft Word. | 1 |
| 3 | Reference Book | Client Side Scripting Language | 1 |

**6.0 Name of team members with Roll No.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.**  **No** | **Roll.**  **No** | **Enrollment No** | **Name** |
| 1 | 03 | 2000160343 | BARDAPURKAR APURVA |
| 2 | 05 | 2000160345 | BUGE TANUJA |
| 3 | 22 | 2000160364 | KAWATHE MADHURA |
| 4 | 23 | 2000160365 | KENDRE KANHOPATRA |

**Part – B**

**Title of Micro-Project**

### “ SIMULATION OF SORTING ”

**1.0 Rationale:**

JavaScript is limited featured client side programming language. JavaScript runs at the client end through the user’s browser without sending messages back and forth to the server. It is widely used by the web developers to do things such as build dynamic webpage respond to events, create interactive forms, validate data that the visitor enters into form, control the browser etc. This project help us to create highly interactive web page using these features.

**2.0 Aim of project:**

Develop website for Simulation Of Sorting.

**3.0 Course Outcomes :**





* Create interactive web pages using program flow control structure.
* Create interactive webpage using functions.
* Create event based web forms using Java script.

**4.0 Literature Review :**

Sorting is a process of ordering or placing a list of elements from a collection in some kind of order. It is nothing but storage of data in sorted order. Sorting can be done in ascending order. It arranges the data in a sequence which makes searching easier.

**Sorting can be performed using several techniques or methods, as follows:**

1. Bubble Sort
2. Insertion Sort
3. Selection Sort
4. Quick Sort
5. Heap Sort

**5.0 Actual Metholodgy Followed :**

This Document Describes Simulation of Sorting .

* 1. Colored representation of step being executed. **Blue:**default . **Yellow:** Being compared **Red:** Identified as in incorrect position and to be moved 1.**Green:** In correct position.
  2. Controls for visualizations 2.1) Speed of visualization (5 speed levels) 2.2) Data size () 2.3) Generation of new data (Randomly generate new data).
  3. Time and Space complexity of algorithm being visualized.

**5.0 Resources Required :**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Name of**  **Resources** | **Specifications** | **Quantity** |
| 1 | Computer | Processor-i5 RAM-8GB  HDD-1TB | 1 |
| 2 | Software | Notepad++,Bracket, Microsoft Word. | 1 |
| 3 | Reference Book | Client Side Scripting Language | 1 |

**8.0 Skill Developed:**

* + 1. Presentation skill
    2. Communication skill
    3. Documentation skill
    4. Team interaction

**9.0 Application of Micro-Project :**

* + Help to understand Sorting ways.
  + Analyze your customer base.
  + Inner Workings Of Sorting Algorithm.

**Code:**

✓ **Html:**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8">**

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

**<title>Sorting Visualizer</title>**

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

**</head>**

**<body>**

**<header>**

**<nav>**

**<div class="array-inputs">**

**<p>Size of the array:</p>**

**<input id="a\_size" type="range" min=20 max=150 step=1 value=80>**

**<p>Speed of the algorithm:</p>**

**<input id="a\_speed" type="range" min=1 max=5 step=1 value=4>**

**<button id="a\_generate">Generate New Array!</button>**

**</div>**

**<div class="header\_right">**

**<p class="nav-heading">Sorting visualizer</p>**

**<div class="algos">**

**<button >Bubble</button>**

**<button >Selection</button>**

**<button >Insertion</button>**

**<button >Merge</button>**

**<button >Quick</button>**

**<button style="margin-right: 0px;">Heap</button>**

**</div>**

**</div>**

**</nav>**

**</header>**

**<section>**

**<div id="Info\_Cont1">**

**<h3>TIME COMPLEXITY</h3>**

**<div class="Complexity\_Containers" id="Time\_Complexity\_Types\_Container">**

**<div class="Pair\_Definitions">**

**<p class="Sub\_Heading">Worst case:</p>**

**<p id="Time\_Worst"></p></div>**

**<div class="Pair\_Definitions">**

**<p class="Sub\_Heading">Average case:</p>**

**<p id="Time\_Average"></p>**

**</div>**

**<div class="Pair\_Definitions">**

**<p class="Sub\_Heading">Best case:</p>**

**<p id="Time\_Best"></p>**

**</div>**

**</div>**

**</div>**

**<div id="array\_container">**

**</div>**

**<div id="Info\_Cont2">**

**<h3>SPACE COMPLEXITY</h3>**

**<div class="Complexity\_Containers" id="Space\_Complexity\_Types\_Container">**

**<div class="Pair\_Definitions">**

**<p class="Sub\_Heading">Worst case:</p>**

**<p id="Space\_Worst"></p>**

**</div>**

**</div>**

**</div>**

**</section>**

**<footer>**

**</footer>**

**<script src="./main.js"></script> <!--This should be at the end of body and should be loaded before sorts.js-->**

**<script src="./visualizations.js"></script> <!--This should be loaded after main.js-->**

**<script src="./bubble\_sort.js"></script>**

**<script src="./selection\_sort.js"></script>**

**<script src="./insertion\_sort.js"></script>**

**<script src="./merge\_sort.js"></script>**

**<script src="./quick\_sort.js"></script>**

**<script src="./heap\_sort.js"></script>**

**</body>**

**</html>**

✓ **CSS**

**$no\_algos: 6;**

**\* { margin: 0px; padding: 0px;**

**font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif;**

**font-size: large; font-weight: bold;**

**text-transform: uppercase;**

**}**

**nav { background-color: #3b3a3a; color: white; display: -ms-grid;**

**display: grid;**

**-ms-grid-columns: 30% 60%; grid-template-columns: 30% 60%; -webkit-column-gap: 10%; column-gap: 10%;**

**padding: 0% 5%;**

**}**

**nav .array-inputs { display: -ms-grid;**

**display: grid;**

**-ms-grid-columns: 60% 40%; grid-template-columns: 60% 40%; place-content: center;**

**padding: 5% 0%;**

**}**

**nav .array-inputs #a\_size {**

**padding: 5% 0%;**

**}**

**nav .array-inputs #a\_speed {**

**padding: 5% 0%;**

**}**

**nav .array-inputs #a\_generate { background-color: transparent; border: none; outline: none;**

**color: white; padding: 4% 0% 0% 0%;**

**padding-left: 20%;**

**}**

**nav .array-inputs #a\_generate:hover {**

**cursor: pointer;**

**}**

**nav .header\_right { display: -webkit-box; display: -ms-flexbox; display: flex; -webkit-box-orient: vertical;**

**-webkit-box-direction: normal; -ms-flex-direction: column;**

**flex-direction: column;**

**}**

**nav .header\_right .nav-heading { display: inline-block; font-size: xxx-large;**

**padding: 3% 0%;**

**}**

**nav .header\_right .algos { display: -ms-grid;**

**display: grid;**

**-ms-grid-columns: (16.66667%)[6]; grid-template-columns: repeat(6, 16.66667%); place-content: center;**

**padding: 2% 0%;**

**}**

**nav .header\_right .algos button { background-color: transparent; border: none; color: white; outline: none; padding: 5% 0%;**

**font-size: x-large;**

**}**

**nav .header\_right .algos button:hover { background-color: blue;**

**cursor: pointer;**

**}**

**nav .header\_right .algos .butt\_locked { background-color: transparent; cursor: pointer;**

**}**

**nav .header\_right .algos .butt\_locked:hover { background-color: transparent;**

**cursor: pointer;**

**}**

**nav .header\_right .algos .butt\_selected { background-color: white; color: green;**

**}**

**nav .header\_right .algos .butt\_selected:hover { background-color: white;**

**cursor: pointer;**

**}**

**nav .header\_right .algos .butt\_unselected:hover { background-color: blue;**

**cursor: pointer;**

**}**

**section { display: -ms-grid;**

**display: grid;**

**-ms-grid-columns: 20% 60% 20%;**

**grid-template-columns: 20% 60% 20%;**

**}**

**section .Complexity\_Containers {**

**margin-top: 20%;**

**}**

**section .Complexity\_Containers .Pair\_Definitions { margin-top: 20%;**

**}**

**section .Complexity\_Containers .Pair\_Definitions p {**

**display: inline;**

**}**

**section .Complexity\_Containers .Pair\_Definitions p.Sub\_Heading { font-size: medium;**

**text-transform: none;**

**}**

**section #Info\_Cont1 {**

**padding: 20% 10%;**

**}**

**section #Info\_Cont1 h3 {**

**text-decoration: underline;**

**}**

**section #array\_container { display: -webkit-box; display: -ms-flexbox; display: flex; width: 100%; height: 70vh;**

**}**

**section #Info\_Cont2 {**

**padding: 20% 10%;**

**}**

**section #Info\_Cont2 h3 {**

**text-decoration: underline;**

**}**

**.hide { display: none;**

**}**

**/\***

**@media (max-width: 480px)//Mobiles**

**{**

**}**

**@media (min-width: 481px) and (max-width: 1250px)//Tablets**

**{**

**}**

**@media (min-width: 1251px)//Laptops and up**

**{**

**}**

✓ **JAVASCRIPT:**

**var speed=1000;**

**inp\_aspeed.addEventListener("input",vis\_speed);**

**function vis\_speed()**

**{**

**var array\_speed=inp\_aspeed.value;**

**switch(parseInt(array\_speed))**

**{**

**case 1: speed=1; break;**

**case 2: speed=10; break;**

**case 3: speed=100; break;**

**case 4: speed=1000;**

**break;**

**case 5: speed=10000;**

**break;**

**}**

**delay\_time=10000/(Math.floor(array\_size/10)\*speed); //Decrease numerator to increase speed.**

**}**

**var delay\_time=10000/(Math.floor(array\_size/10)\*speed); //Decrease numerator to increase speed. var c\_delay=0;//This is updated ov every div change so that visualization is visible.**

**function div\_update(cont,height,color)**

**{**

**window.setTimeout(function(){**

**cont.style=" margin:0% " + margin\_size + "%; width:" + (100/array\_size-(2\*margin\_size)) + "%; height:" + height + "%; background-color:" + color + ";";**

**},c\_delay+=delay\_time);**

**}**

**function enable\_buttons()**

**{**

**window.setTimeout(function(){**

**for(var i=0;i<butts\_algos.length;i++)**

**{**

**butts\_algos[i].classList=[];**

**butts\_algos[i].classList.add("butt\_unselected");**

**butts\_algos[i].disabled=false; inp\_as.disabled=false; inp\_gen.disabled=false;**

**inp\_aspeed.disabled=false;**

**}**

**},c\_delay+=delay\_time);**

**}**

**function Bubble()**

**{**

**//Setting Time complexities**

**document.getElementById("Time\_Worst").innerText="O(N^2)"; document.getElementById("Time\_Average").innerText="Θ(N^2)"; document.getElementById("Time\_Best").innerText="Ω(N)";**

**//Setting Space complexity**

**document.getElementById("Space\_Worst").innerText="O(1)";**

**c\_delay=0;**

**for(var i=0;i<array\_size-1;i++)**

**{**

**for(var j=0;j<array\_size-i-1;j++)**

**{**

**div\_update(divs[j],div\_sizes[j],"yellow");//Color update**

**if(div\_sizes[j]>div\_sizes[j+1])**

**{**

**div\_update(divs[j],div\_sizes[j], "red");//Color update div\_update(divs[j+1],div\_sizes[j+1], "red");//Color update**

**var temp=div\_sizes[j]; div\_sizes[j]=div\_sizes[j+1];**

**div\_sizes[j+1]=temp;**

**div\_update(divs[j],div\_sizes[j], "red");//Height update**

**div\_update(divs[j+1],div\_sizes[j+1], "red");//Height update**

**}**

**div\_update(divs[j],div\_sizes[j], "blue");//Color updat**

**}**

**div\_update(divs[j],div\_sizes[j], "green");//Color update**

**}**

**div\_update(divs[0],div\_sizes[0], "green");//Color update**

**enable\_buttons();**

**}**

**function Heap()**

**{**

**//Setting Time complexities**

**document.getElementById("Time\_Worst").innerText="O(N log N)"; document.getElementById("Time\_Average").innerText="Θ(N log N)"; document.getElementById("Time\_Best").innerText="Ω(N log N)";**

**//Setting Space complexity**

**document.getElementById("Space\_Worst").innerText="O(1)";**

**c\_delay=0;**

**heap\_sort();**

**enable\_buttons();**

**}**

**function swap(i,j)**

**{**

**div\_update(divs[i],div\_sizes[i],"red");//Color update div\_update(divs[j],div\_sizes[j],"red");//Color update**

**var temp=div\_sizes[i]; div\_sizes[i]=div\_sizes[j];**

**div\_sizes[j]=temp;**

**div\_update(divs[i],div\_sizes[i],"red");//Height update div\_update(divs[j],div\_sizes[j],"red");//Height update**

**div\_update(divs[i],div\_sizes[i],"blue");//Color update div\_update(divs[j],div\_sizes[j],"blue");//Color update**

**}**

**function max\_heapify(n,i)**

**{ var largest=i; var l=2\*i+1; var r=2\*i+2;**

**if(l<n && div\_sizes[l]>div\_sizes[largest])**

**{**

**if(largest!=i)**

**{**

**div\_update(divs[largest],div\_sizes[largest],"blue");//Color update**

**}**

**largest=l;**

**div\_update(divs[largest],div\_sizes[largest],"red");//Color update**

**}**

**if(r<n && div\_sizes[r]>div\_sizes[largest])**

**{**

**if(largest!=i)**

**{**

**div\_update(divs[largest],div\_sizes[largest],"blue");//Color update**

**}**

**largest=r;**

**div\_update(divs[largest],div\_sizes[largest],"red");//Color update**

**}**

**if(largest!=i)**

**{**

**swap(i,largest);**

**max\_heapify(n,largest);**

**}**

**}**

**function heap\_sort()**

**{**

**for(var i=Math.floor(array\_size/2)-1;i>=0;i--)**

**{**

**max\_heapify(array\_size,i);**

**}**

**for(var i=array\_size-1;i>0;i--)**

**{**

**swap(0,i);**

**div\_update(divs[i],div\_sizes[i],"green");//Color update**

**div\_update(divs[i],div\_sizes[i],"yellow");//Color update**

**max\_heapify(i,0);**

**div\_update(divs[i],div\_sizes[i],"blue");//Color update**

**div\_update(divs[i],div\_sizes[i],"green");//Color update**

**}**

**div\_update(divs[i],div\_sizes[i],"green");//Color update**

**}**

**function Insertion()**

**{**

**//Setting Time complexities**

**document.getElementById("Time\_Worst").innerText="O(N^2)"; document.getElementById("Time\_Average").innerText="Θ(N^2)";**

**document.getElementById("Time\_Best").innerText="Ω(N)";**

**//Setting Space complexity**

**document.getElementById("Space\_Worst").innerText="O(1)";**

**c\_delay=0;**

**for(var j=0;j<array\_size;j++)**

**{**

**div\_update(divs[j],div\_sizes[j],"yellow");//Color update**

**var key= div\_sizes[j];**

**var i=j-1;**

**while(i>=0 && div\_sizes[i]>key)**

**{**

**div\_update(divs[i],div\_sizes[i],"red");//Color update**

**div\_update(divs[i+1],div\_sizes[i+1],"red");//Color update**

**div\_sizes[i+1]=div\_sizes[i];**

**div\_update(divs[i],div\_sizes[i],"red");//Height update**

**div\_update(divs[i+1],div\_sizes[i+1],"red");//Height update**

**div\_update(divs[i],div\_sizes[i],"blue");//Color update if(i==(j-1))**

**{**

**div\_update(divs[i+1],div\_sizes[i+1],"yellow");//Color update**

**} else**

**{**

**div\_update(divs[i+1],div\_sizes[i+1],"blue");//Color update**

**} i-=1;**

**}**

**div\_sizes[i+1]=key;**

**for(var t=0;t<j;t++)**

**{**

**div\_update(divs[t],div\_sizes[t],"green");//Color update**

**}**

**}**

**div\_update(divs[j-1],div\_sizes[j-1],"green");//Color update**

**enable\_buttons();**

**}**

**var inp\_as=document.getElementById('a\_size'),array\_size=inp\_as.value; var inp\_gen=document.getElementById("a\_generate"); var inp\_aspeed=document.getElementById("a\_speed");**

**//var array\_speed=document.getElementById('a\_speed').value;**

**var butts\_algos=document.querySelectorAll(".algos button");**

**var div\_sizes=[];**

**var divs=[]; var margin\_size;**

**var cont=document.getElementById("array\_container"); cont.style="flex-direction:row";**

**//Array generation and updation.**

**inp\_gen.addEventListener("click",generate\_array); inp\_as.addEventListener("input",update\_array\_size);**

**function generate\_array()**

**{**

**cont.innerHTML="";**

**for(var i=0;i<array\_size;i++)**

**{**

**div\_sizes[i]=Math.floor(Math.random() \* 0.5\*(inp\_as.max - inp\_as.min) ) + 10; divs[i]=document.createElement("div");**

**cont.appendChild(divs[i]); margin\_size=0.1;**

**divs[i].style=" margin:0% " + margin\_size + "%; background-color:blue; width:" + (100/array\_size-(2\*margin\_size)) +**

**"%; height:" + (div\_sizes[i]) + "%;";**

**}**

**}**

**function update\_array\_size()**

**{**

**array\_size=inp\_as.value;**

**generate\_array();**

**}**

**window.onload=update\_array\_size();**

**//Running the appropriate algorithm. for(var i=0;i<butts\_algos.length;i++)**

**{**

**butts\_algos[i].addEventListener("click",runalgo);**

**}**

**function disable\_buttons()**

**{**

**for(var i=0;i<butts\_algos.length;i++)**

**{**

**butts\_algos[i].classList=[];**

**butts\_algos[i].classList.add("butt\_locked");**

**butts\_algos[i].disabled=true; inp\_as.disabled=true; inp\_gen.disabled=true;**

**inp\_aspeed.disabled=true;**

**}**

**}**

**function runalgo()**

**{**

**disable\_buttons();**

**this.classList.add("butt\_selected"); switch(this.innerHTML)**

**{**

**case "Bubble":Bubble();**

**break;**

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

**break;**

**case "Insertion":Insertion();**

**break;**

**case "Merge":Merge();**

**break;**

**case "Quick":Quick();**

**break;**

**case "Heap":Heap();**

**break;**

**}**

**}**

**function Merge()**

**{**

**//Setting Time complexities**

**document.getElementById("Time\_Worst").innerText="O(N log N)"; document.getElementById("Time\_Average").innerText="Θ(N log N)"; document.getElementById("Time\_Best").innerText="Ω(N log N)";**

**//Setting Space complexity**

**document.getElementById("Space\_Worst").innerText="O(N)";**

**c\_delay=0;**

**merge\_partition(0,array\_size-1);**

**enable\_buttons();**

**}**

**function merge\_sort(start,mid,end)**

**{**

**var p=start,q=mid+1;**

**var Arr=[],k=0;**

**for(var i=start; i<=end; i++)**

**{**

**if(p>mid)**

**{**

**Arr[k++]=div\_sizes[q++];**

**div\_update(divs[q-1],div\_sizes[q-1],"red");//Color update**

**}**

**else if(q>end)**

**{**

**Arr[k++]=div\_sizes[p++];**

**div\_update(divs[p-1],div\_sizes[p-1],"red");//Color update**

**}**

**else if(div\_sizes[p]<div\_sizes[q])**

**{**

**Arr[k++]=div\_sizes[p++];**

**div\_update(divs[p-1],div\_sizes[p-1],"red");//Color update**

**} else**

**{**

**Arr[k++]=div\_sizes[q++];**

**div\_update(divs[q-1],div\_sizes[q-1],"red");//Color update**

**} }**

**for(var t=0;t<k;t++)**

**{**

**div\_sizes[start++]=Arr[t];**

**div\_update(divs[start-1],div\_sizes[start-1],"green");//Color update**

**}** }

**function merge\_partition(start,end)**

**{**

**if(start < end)**

**{**

**var mid=Math.floor((start + end) / 2);**

**div\_update(divs[mid],div\_sizes[mid],"yellow");//Color update**

**merge\_partition(start,mid); merge\_partition(mid+1,end);**

**merge\_sort(start,mid,end);**

**}**

**}**

**function Quick()**

**{**

**//Setting Time complexities**

**document.getElementById("Time\_Worst").innerText="O(N^2)"; document.getElementById("Time\_Average").innerText="Θ(N log N)"; document.getElementById("Time\_Best").innerText="Ω(N log N)";**

**//Setting Space complexity**

**document.getElementById("Space\_Worst").innerText="O(log N)"; c\_delay=0;**

**quick\_sort(0,array\_size-1);**

**enable\_buttons();**

**}**

**function quick\_partition (start, end)**

**{**

**var i = start + 1; var piv = div\_sizes[start] ;//make the first element as pivot element. div\_update(divs[start],div\_sizes[start],"yellow");//Color update**

**for(var j =start + 1; j <= end ; j++ )**

**{**

**//re-arrange the array by putting elements which are less than pivot on one side and which are greater that on other.**

**if (div\_sizes[ j ] < piv)**

**{**

**div\_update(divs[j],div\_sizes[j],"yellow");//Color update**

**div\_update(divs[i],div\_sizes[i],"red");//Color update div\_update(divs[j],div\_sizes[j],"red");//Color update**

**var temp=div\_sizes[i]; div\_sizes[i]=div\_sizes[j];**

**div\_sizes[j]=temp;**

**div\_update(divs[i],div\_sizes[i],"red");//Height update div\_update(divs[j],div\_sizes[j],"red");//Height update**

**div\_update(divs[i],div\_sizes[i],"blue");//Height update div\_update(divs[j],div\_sizes[j],"blue");//Height update**

**i += 1;**

**}**

**}**

**div\_update(divs[start],div\_sizes[start],"red");//Color update div\_update(divs[i-1],div\_sizes[i-1],"red");//Color update**

**var temp=div\_sizes[start];//put the pivot element in its proper place.**

**div\_sizes[start]=div\_sizes[i-1]; div\_sizes[i-1]=temp;**

**div\_update(divs[start],div\_sizes[start],"red");//Height update div\_update(divs[i-1],div\_sizes[i-1],"red");//Height update**

**for(var t=start;t<=i;t++)**

**{**

**div\_update(divs[t],div\_sizes[t],"green");//Color update**

**}**

**return i-1;//return the position of the pivot**

**}**

**function quick\_sort (start, end )**

**{**

**if( start < end )**

**{**

**//stores the position of pivot element var piv\_pos = quick\_partition (start, end ) ;**

**quick\_sort (start, piv\_pos -1);//sorts the left side of pivot. quick\_sort (piv\_pos +1, end) ;//sorts the right side of pivot.**

**}**

**}**

**function Selection\_sort()**

**{**

**//Setting Time complexities**

**document.getElementById("Time\_Worst").innerText="O(N^2)"; document.getElementById("Time\_Average").innerText="Θ(N^2)"; document.getElementById("Time\_Best").innerText="Ω(N^2)";**

**//Setting Space complexity**

**document.getElementById("Space\_Worst").innerText="O(1)";**

**c\_delay=0;**

**for(var i=0;i<array\_size-1;i++)**

**{**

**div\_update(divs[i],div\_sizes[i],"red");//Color update index\_min=i; for(var j=i+1;j<array\_size;j++)**

**{**

**div\_update(divs[j],div\_sizes[j],"yellow");//Color update**

**if(div\_sizes[j]<div\_sizes[index\_min])**

**{**

**if(index\_min!=i)**

**{**

**div\_update(divs[index\_min],div\_sizes[index\_min],"blue");//Color update**

**}**

**index\_min=j;**

**div\_update(divs[index\_min],div\_sizes[index\_min],"red");//Color update**

**} else**

**{**

**div\_update(divs[j],div\_sizes[j],"blue");//Color update**

**}**

**}**

**if(index\_min!=i)**

**{**

**var temp=div\_sizes[index\_min]; div\_sizes[index\_min]=div\_sizes[i];**

**div\_sizes[i]=temp;**

**div\_update(divs[index\_min],div\_sizes[index\_min],"red");//Height update**

**div\_update(divs[i],div\_sizes[i],"red");//Height update**

**div\_update(divs[index\_min],div\_sizes[index\_min],"blue");//Color update**

**}**

**div\_update(divs[i],div\_sizes[i],"green");//Color update**

**}**

#### div\_update(divs[i],div\_sizes[i],"green");//Color update

**enable\_buttons();**

**}**

 **OUTPUT**

