A Project report on

SORTING ALGO TRACKER

A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

**Bachelor of Technology**

**In**

**Computer Science and Engineering**

Submitted by

A.BHAGYA SREE

20H51A0557

I.SREE ANVITA

20H51A0566

M.PRIYANKA

20H51A0568

Under the esteemed guidance of

P.Sravanthi

(Assistant professor)



Department of Computer Science and Engineering

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

(An Autonomous Institution, Approved by AICTE, Affiliated to JNTUH, NAAC ’A+’)

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

2020- 2024

### CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



#### CERTIFICATE

This is to certify that the Project report entitled " sorting algo tracker " being

submitted by A.Bhagya sree (20H51A0557), I.Sree Anvita (20H51A0566), M.Priyanka (20H51A0568), in partial fulfillment for the award of Bachelor of Technology in Computer Science and Engineering is a record of bonafide work carried out his/her under my guidance and supervision.

The results embodied in this project report have not been submitted to

any other University or Institute for the award of any Degree.

**P.Sravanti Dr.S.Siva Skandha**

**Associative Professor**  **Associate** **Professor and Head CSE**

**Dept. of CSE Dept. of CSE**

**Acknowledgment**

With great pleasure, I want to take this opportunity to express my heartfelt gratitude to all the people who helped in making this project work a grand success.

I am grateful to **P.Sravanti,** Associative Professor, Dept of Computer Science and Engineering for his valuable suggestions and guidance during the execution of this project work.

I would like to thank **Dr.S.Siva Skandha** Head of the Department of Computer Science and Engineering, for his moral support throughout the period of my study in CMRCET.

I am highly indebted to **Dr. V A Narayana**, Principal CMRCET for giving permission to carry out this project in a successful and fruitful way.

I would like to thank the Teaching & Non- teaching staff of the Department of Computer Science and Engineering for their co-operation

Finally, I express my sincere thanks to **Mr. Ch. Gopal Reddy**, Secretary, CMR Group of Institutions, for his continuous care. I sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project work.

A.Bhagya Sree

20H51A0557

I.Sree Anvita

20H51A0566

M.Priyanka

20H51A0568

**TABLE OF CONTENTS**

**CHAPTER**

**NO. TITLE PAGE NO.**

**LIST OF FIGURES** i

**ABSTRACT**  1

**1 INTRODUCTION**  2

**2 BACKGROUND WORK** 3

2.1 EXISTING SYSTEM 3

2.1.1 MANUALLY 3

2.1.2 DISADVANTAGES 4

**3 PROPOSED SYSTEM** 5

3.1 INTRODUCTION 6

3.2 IMPLEMENTATION 8

**4 DESIGNING** 9

4.1 UML DIAGRAMS 9

4.2 USE CASE DIAGRAMS 9

4.3 CODE DESIGNING 13

**5 RESULTS AND DISCISSION** 14

**6 CONCLUSION AND FUTUREWORK** 15

**7 REFERENCES** 16

## **ABSTRACT**

* Learning about sorting algorithms has become a tough task, for a student they often failed to understand the core idea of a particular algorithm maybe because they are unable to visualize how they work.
* This web application makes everyone to understand how algorithms works through visualizations.
* This project is built by using HTML,CSS,JAVASCRIPT.
* This sorting algos is a very simple UI which allows user to select the sort algorithm, select the array size and the speed of visualization.
* To visualize four sorting algorithms, a web-based animation application was constructed.
* A visualization of data is implemented as a bar graph, after which a data sorting and algorithm may be applied. The resulting animation
* is then performed either automatically or by the user, who then sets their own pace. This is a research on the computer science curriculum's approach to learning algorithms. The experiment featured a presentation and a survey, both of which asked students questions which may illustrate improvements in algorithm comprehension. These findings and reactions are catalogued in this document and compared to earlier investigations

CMRCET B.Tech (CSE) Page No 1

CHAPTER 1 : INTRODUCTION

Nowadays sorting algorithms are widely used in computer software. For example, if you open file explorer on your PC, you may see files sorted in different ways. Searching in sorted data is more efficient than in not sorted ones. Students of computer science start learning different algorithms in the first year of studies and sorting algorithms are among them. Since I faced the problems of sorting during the course of algorithm design in the first year of my studies, there is an understanding that the visual representation is a vital part of the studying process. During working on the is it was very exciting to learn different technic sorting

algorithms into the depth.

The main goal of the thesis was to create a program which would serve as a tool

for understanding how most known sorting algorithms work. There was an

attempt to make the best possible user experience. The demonstration software is

made in a user-friendly and easy-to-use style. To gain maximal benefit from

learning you can try each sorting algorithm on your data. The text of the

the is describes principles of the most known sorting algorithms which are

demonstrated in the computer program.

It might be used as a source for learning algorithms by students. Also, the

program might be easily used as a demonstration by lecturers and tutors during

classes.

Besides, there is programmer documentation and user guide to the provided

software. Readers of this text are expected to have some programming experience

to know basic data structures such as arrays, lists, trees and understand recursive

procedures. Also, knowledge of some simple algorithms and their implementations

could be helpful. In order to understand the topic better, knowledge of linear

algebra and calculus is involved.



CMRCET B.Tech (CSE) Page No 2

CHAPTER 2 : BACKGROUND WORK

2.1 Existing Systems:

2.1.1 Manually:

Students learn these sorting algorithms through textbooks and from faculty. firstly they learn about an algorithm.

Using references of Wikipedia and videos students could learn about this algos.

**Algorithms:**

Analysis and design of algorithms is a great challenge for both computer and information science students.

Algorithm visualization uses computer graphics or multimedia to show the actions of an algorithm step by step. In this research, the algorithm visualization is used to help students understand the concept of selection sort algorithm and creating coding to visualize it. When it comes to [computer programming](https://www.technotification.com/category/coding), algorithms work in a similar manner. In layman’s language, an algorithm can be defined as a step-by-step procedure for accomplishing a task. In the world of programming, an algorithm is a well-structured computational procedure that takes some values as input and some values as output. Algorithms give us the most ideal option for accomplishing a task. Here is some importance of algorithms in computer programming.

**To improve the efficiency of a computer program:**

In programming, there are different ways of solving a problem. However, the efficiency of the methods available varies. Some methods are well suited to give more accurate answers than others. Algorithms are used to find the best possible way of solving a problem. In doing so they improve the efficiency of a program.

**Types of Algorithms:**

Here are the most common type of algorithms:

CMRCET B.Tech (CSE) Page No 3

* Searching algorithms
* Pathfinding algorithm
* Sorting algorithms
* Compression algorithms
* Tree and graph-based algorithms
* Pattern matching algorithm among many others

**2.2 Disadvantages in existing system:**

* Unable to trace the algorithms because it consumes lot of time
* Students tend to lose intrest and patience while working on large set of inputs.
* Through manual methods students face issue in visualization and this leads to low efficiency.

****

CMRCET B.Tech (CSE) Page No 4

CHAPTER 3 :

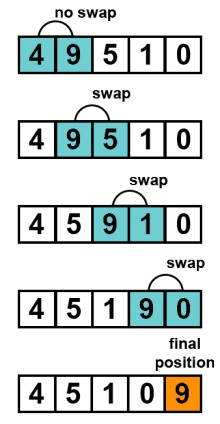
PROPOSED SOLUTION:

3.1 INTRODUCTION:

Our proposed solution is to develop a very simple UI which is useful to understand the sorting algorithms in an effective way .

Our UI includes the following sorting algorithms through visualizations

1.BUBBLE SORT:

 Bubble Sort is based on the idea of exchanging two adjacent elements if they have the wrong order. The algorithm works stepping through all elements from left to right, so the largest elements tend to move or "bubble" to the right .That is why the algorithm is called Bubble Sort. Now we are going to the details. Let us have an unsorted array.

The algorithm does iterations through the unsorted part which is the whole array at the beginning. And with each iteration through the array the range of inspected items is decreased by one till only two elements left. After this two elements are compared and possibly swapped, the array is considered as sorted. Bubble Sort complexity is Θ(n 2 ).

2.SELECTION SORT:

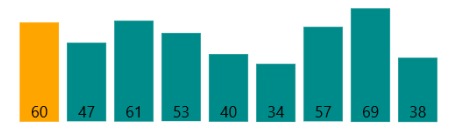
Selection Sort algorithm is based on the repeated selection. Here we consider finding minimal key from the unsorted part and swapping it with the first unsorted key. As well as in the Insertion Sort, sorted part grows from the beginning of the sequence. Assume an array of items to sort. At the beginning of the sorting process unsorted part is represented by the whole array. Then, the first item of the unsorted part is set as the smallest item and is compared with the follow-up elements. When smaller item is found, it is set as a new smallest key. After the end of the array is reached the smallest item is swapped with the first element of the unsorted part and it becomes the sorted part of the array. This step is repeated till the array is sorted. Complexity of this sorting algorithm is Θ(n 2 ).

3.INSERTION SORT:

Insertion Sort algorithm has a simple idea. Assume an array with items to be sorted. We divide the array into two parts: sorted one and unsorted one. At the beginning sorted part consists of the first element .Then, for each item that we have in the unsorted part, we take element and insert it into the right place among the sorted items.

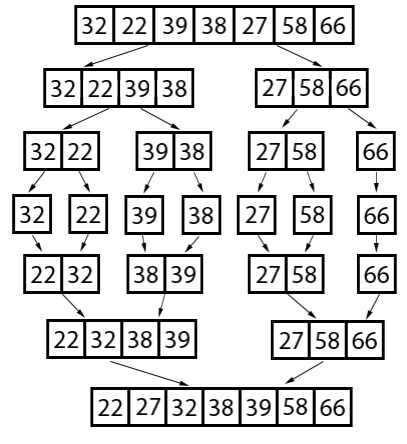
CMRCET B.Tech (CSE) Page No 5

In order to insert element into the right place in the sorted part, we compare selected item from the unsorted part with each item from the sorted part in the direction from right to left. Comparing continues until smaller or equal element is found or no elements to compare left. After each comparison, if current item in the sorted part is greater, we move that current item one position right. Finally, when the right position is found, we insert an item into the sorted part. Complexity of Insertion Sort is Θ(n 2 ).



MERGE SORT:

Merge Sort as well as Quick Sort is an algorithm of type "divide and conquer". Its logic is simple: divide data into two parts, sort the left part, sort the right part, then "merge" the parts back. The algorithm works by the recursive application itself on the unsorted parts. In the beginning, it selects the middle item, which becomes the rightmost element of the left part. Then, it recursively sorts both parts. Finally, the algorithm "merges" two sorted parts. Merging procedure itself takes items from each of two sorted parts one by one, compares them and moves the smallest to the output, repeats the previous step. Merge Sort complexity is Θ(n log n)



CMRCET B.Tech (CSE) Page No 6

3.2 IMPLEMENTATION:

The use of HTML5 (Hypertext Markup Language 5), JavaScript, and CSS combine to form this project's implementation (Cascading Style Sheets). There is only one project file which is an HTML file and contains the code. The only additional piece of code added to the main HTML file is the .m4a sound files to support the sound animation functionality (which are saved as .m4a files). As of now, I only did extensive testing using the Mozilla Firefox browser, and it's the browser of choice in this context. However, tests done quickly revealed the possibility of Google Chrome and Safari integration.

This software uses both object-oriented and functional programming paradigms in how it organizes the code. Before the final phase of development, the design was almost completely functional, where only three objects were used: one to control the canvas that displayed the animation, another to represent a piece of data, or “bar” object (blue rectangle with dynamically changing height and position), and a final one to represent the positions that each bar moved to, or “pos” objects. Although this incorporated several function calls, some instance variables and Boolean values were utilized to keep track of the algorithm picked and when to animation, but this led to a greatly integrated mass of code that was difficult to maintain. Several big re factorings later, the code has now taken on the form of a Model-View-Controller Architecture. Although, because of its functional character, it possesses a multitude of individualized functions that alter the instance variables and Boolean values, which means it has a multitude of functions that directly alter the View and Controller. The major module in the HTML code between the <script> and </script> tags is known as the global scope. Everything within the framework is able to access the aforementioned variables and methods.

a) The View

There are three items on the view: the sortArea, the bar, and the position. These objects operate within the container defined by the <script> and </script> tags in the .html file. This function's space is sometimes referred to as the "main" function, the first function invoked in a program.

It is the sortArea that keeps the bars up to date using a timer, while at the same time generating the bar graph. As a result, whenever "Step" is invoked, the bar values are updated depending on the steps array (discussed later). In the sortArea, after every second iteration of the timer, the rectangles will be redrawn with varied heights that represent the new values. The bars change sixty times per second, so when the “Step” button is selected, the change is instantaneous. In the sortArea, the bar object represents each piece of data. The statement encompasses all of the aspects of color, value, location, height, and sound. While having a distinct array named bars for the current bars in the bar graph helps preserve attributes such as the total number of bars (total value) independent of other characteristics, it is simple to update any or all of the attributes by iterating over the array as necessary.

CMRCET B.Tech (CSE) Page No 7

b) The Model:

The model is made up of one item, known as the sorter. This object houses the algorithm's code divided into methods.

Sort method centralizes on an integer constant and uses it to order the algorithm's possible algorithms.

This object is directly controlled by the four sorting algorithms shown on the user interface as "Selection Sort, Bubble Sort, Insertion Sort, and Merge/Insertion Sort." The sort algorithm method is invoked as the user selects a sorting technique and clicks on one of the sort algorithm buttons.

Then, when the algorithm sorts the data, a trace is created.

The steps array, which contains all the movements in the animation, is a two dimensional integer array that is available to methods on the web user interface

To complete the algorithm's walkthrough, the View will cycle through the data and update the bars in the bar graph to show how the algorithm calculated the steps it took.

It's important to note that if the algorithm generated a change in the position of a piece of data, the steps are merely recorded.

c) The Controller

Every web page on the web has at least one Controller. The buttons have been programmed in HTML and lead the browser to execute a JavaScript procedure upon clicking.

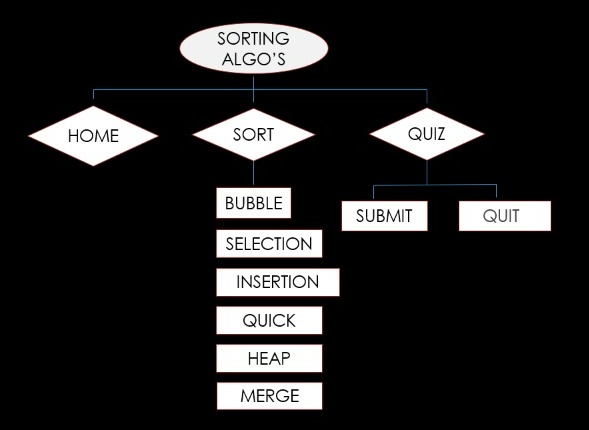
The CSS is used for making the buttons look attractive, arranging them, choosing their colors, and applying visual effects.

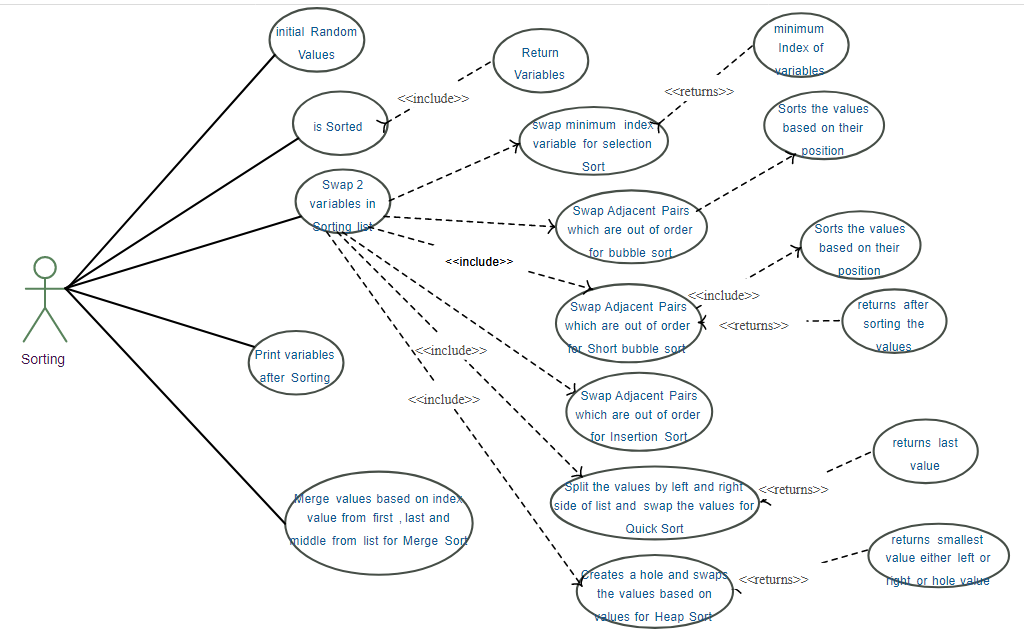
The methods alter the Model's state, prepare it for the update, and then apply the modifications to the View. Four groups of buttons are distinguished

CMRCET B.Tech (CSE) Page No 8

CHAPTER 4: DESIGING

4.1:UML DIAGRAM



4.2 USE CASE: 

CMRCET B.Tech (CSE) Page No 9

4.3 CODE DESIGNING

HTML:

<!DOCTYPE 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="https://pro.fontawesome.com/releases/v5.10.0/css/all.css" integrity="sha384-AYmEC3Yw5cVb3ZcuHtOA93w35dYTsvhLPVnYs9eStHfGJvOvKxVfELGroGkvsg+p" crossorigin="anonymous"/>

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

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

<link href="https://fonts.googleapis.com/css?family=Macondo" rel="stylesheet">

</head>

<body>

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

<a class="navbar-brand" href="index.html">SORTING ALGO TRACKER</a>

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

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

</button>

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

<ul class="navbar-nav mr-auto">

<li class="nav-item active">

<a class="nav-link" href="index.html">Home <span class="sr-only">(current)</span></a>

</li>

CMRCET B.Tech (CSE) Page No 10

<li class="nav-item">

<a class="nav-link" href="sort.html">Sort</a>

</li>

<li class="nav-item">

<a class="nav-link" href="quiz.html">Quiz</a>

</li>

<li class="nav-item">

<a class="nav-link" href="complexities.html">

</a>

</li>

</ul>

<form class="form-inline my-2 my-lg-0">

<input class="form-control mr-sm-2" type="search" placeholder="Search" aria-label="Search">

<button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>

</form>

</div>

</nav>

<div class="container mt-5">

<h1 >sorting algo tracker</h1>

<h3 class="display-5">Say education to edutainment !</h3>

<p class="lead">This is educational startup,which focus on new techniques of learning. Visualization is very helpful in learning. Its easy to

understand the sorting algorithms. We have also included the quiz. So lets explore knowledge.

</p>

</div>

<footer class="bg-dark text-center text-white fixed-bottom">

CMRCET B.Tech (CSE) Page No 11

<!-- Grid container -->

<div class="container p-4 pb-0">

<!-- Section: Social media -->

<section class="mb-4">

<!-- Facebook -->

<a class="btn btn-outline-light btn-floating m-1" href="#!" role="button"

><i class="fab fa-facebook-f"></i

></a>

<!-- Twitter -->

<a class="btn btn-outline-light btn-floating m-1" href="#!" role="button"

><i class="fab fa-twitter"></i

></a>

<!-- Google -->

<a class="btn btn-outline-light btn-floating m-1" href="#!" role="button"

><i class="fab fa-google"></i

></a>

<!-- Instagram -->

<a class="btn btn-outline-light btn-floating m-1" href="#!" role="button"

><i class="fab fa-instagram"></i

></a>

<!-- Linkedin -->

<a class="btn btn-outline-light btn-floating m-1" href="https://www.linkedin.com/in/saurabh-jejurkar-b80042195/" role="button"

><i class="fab fa-linkedin-in"></i

></a>

CMRCET B.Tech (CSE) Page No 12

<!-- Github -->

<a class="btn btn-outline-light btn-floating m-1" href="https://github.com/Saurabh641444/"

role="button"

><i class="fab fa-github"></i

></a>

</section>

<!-- Section: Social media -->

</div>

<!-- Grid container -->

<!-- Copyright -->

<div class="text-center p-3" style="background-color: rgba(0, 0, 0, 0.2);">

© 2021 Copyright:

<a class="text-white" href="">Saurabh Jejurkar</a>

</div>

<!-- Copyright -->

</footer>

</body>

</html>

CMRCET B.Tech (CSE) Page No 13

CHAPTER 5:

RESULT AND DISCUSSION

Sorting algo tracker is successfully implemented and working fine. It is useful for many of the people who believes that understanding of sorting algorithms is difficult.

Start by arranging the data, and then pick the visualization algorithm to use. Algorithm buttons provide sorting of data as it arrives on the interface. Asking to specify the ordering of elements takes precedence because when the algorithm has completed running the initialization process, the interface is now showing a new ordering, while the code has already completed running the initialization with the prior data set.

A shortcoming of the animation is that it does not provide comparisons of the data's motions that result in such movements. Selection Sort's performance advantage over the other sorting algorithms is due to the fact that there are O(n) swaps, which eliminates superfluous computer movements. Comparing the data produces a runtime complexity of O(n squared) (the slowest overall). In response to question 5, where students were asked for input and thoughts, another student stated that Merge Sort is the best of the four kinds. The average runtime of Merge Sort is O(n log2 n), which is the best average runtime among all sorting algorithms. Integrating visualization of comparisons as well as motions would help fix this. A good technique to accomplish this is to use an algorithm that highlights the bars in red when it is examining data, requiring additional time in the animation. The following sorting algorithms, Selection Sort and Bubble Sort, would require a considerable amount of comparisons in order to finish.

Features of sorting algo tracker

• Ability to modify the animation's pace

• Using visual feedback while clicking the buttons on the user interface to keep them selected

• By creating a box-like split for Merge Sort, you'll be able to more clearly identify the split phases.

• Colors should be used to show whether or not anything is being compared.

CMRCET B.Tech (CSE) Page No 14

CHAPTER 6:

CONCLUSION AND FUTURE WORK:

By developing sorting algo tracker web application we came to overcome the problem faced by many people who found that understanding of sorting algorithms is tough. This web page will make us understand the sorting algorithms through visualisation in a better way.

This web-based animation tool for viewing the following sorting algorithms functions in great part because of all the time and effort that I invested into it. In spite of its memory overhead, the feedback given to it was mostly good from the students that worked with it. This is consistent with my prior research, which revealed that there was no substantial difference in learning the content. What I do agree with totally is the attitude that holds there is a great need to investigate and produce animated presentations to enhance education in the classroom. Overall, I am not concerned that a large rework to a different language will be required soon because JavaScript is still one of the most popular web languages.

FUTURE WORK

* Including pause and restart buttons.
* Adding control panel
* Helpful for beginners
* easy to understand
* Search engine optimization

CMRCET B.Tech (CSE) Page No 15

CHAPTER 7:

REFERENCES:

* <https://www.geeksforgeeks.org/sorting-algorithms/>
* <https://www.programiz.com/dsa/sorting-algorithm>
* <https://www.javatpoint.com/time-complexity-of-sorting-algorithms>
* <https://www.w3schools.com/>

CMRCET B.Tech (CSE) Page No 16

A Project report on

SORTING ALGO TRACKER

A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

**Bachelor of Technology**

**In**

**Computer Science and Engineering**

Submitted by

20H51A0557

A.Bhagya sree

20H51A0566

I.Sree Anvita

20H51A0568

M.Priyanka

Under the esteemed guidance of

(P.Sravanti)

(Associative Professor)



### Department of Computer Science and Engineering

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

(An Autonomous Institution, Approved by AICTE, Affiliated to JNTUH, NAAC ’A+’)

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

2020- 2024

### CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



#### CERTIFICATE

This is to certify that the Project report entitled " sorting algo tracker " being

submitted by A.Bhagya sree (20H51A0557), I.Sree Anvita (20H51A0566), M.Priyanka (20H51A0568), in partial fulfillment for the award of Bachelor of Technology in Computer Science and Engineering is a record of bonafide work carried out his/her under my guidance and supervision.

The results embodied in this project report have not been submitted to

any other University or Institute for the award of any Degree.

**P.Sravanti Dr.S.Siva Skandha**

**Associative Professor Associate Professor and Head CSE**

**Dept. of CSE Dept. of CSE**