|  |
| --- |
| **Data Structures and Algorithms** |
| **​​Utopian City​** |
| **Course Project Report** |

|  |
| --- |
| **School of Computer Science and Engineering**  **2022-23** |

**Contents**

|  |  |
| --- | --- |
| **Si. No.** | **Topics** |
| 1. | Course and Team Details |
| 2. | Introduction |
| 3. | Problem Definition |
| 4. | Functionalities |
| 5. | Project Tools |
| 6. | Learning and Takeaway |
| 7. | References |

**1. Course and Team Details**

**1.1 Course details**

|  |  |
| --- | --- |
| **Course Name** | Data Structures and Algorithms  (Theory and Lab) |
| **Course Code** | 20ECSC205 and 19ECSP201 |
| **Semester** | III |
| **Division** | A |
| **Year** | 2022-23 |
| **Instructor** | Prakash Hegade |

**1.2 Team Details**

|  |  |  |
| --- | --- | --- |
| **Si. No.** | **Roll No.** | **Name** |
| 1. | 134 | Karthik R Khatavkar |
| 2. | 135 | Avaneesh Lad |
| 3. | 138 | Apeksha M |
| 4. | 139 | Priyanka B M |

**2. Introduction**

Algorithms and data structures are the fundamental units of computer science and programming. They are employed to efficiently organise, process, and store data as well as to address challenging issues. The goal of this course and lab is to give students a broad understanding of basic data structures and algorithms, as well as how to build and analyse them.

The course covers a variety of data structures along with methods for designing algorithms, including divide-and-conquer, dynamic programming, and greedy algorithms. The course also covers fundamental analysis methods for evaluating the effectiveness and complexity of algorithms, such as Big O notation. Arrays, linked lists, stacks, queues, trees, graphs, sorting, searching, and algorithm design techniques are some of the subjects covered in the course. The purpose of the lab sessions is to provide participants hands-on experience testing and implementing various data structures and algorithms.

Implementing and experimenting with the data structures and algorithms presented in the course can be done in the lab sessions. Students will benefit from the lab work by developing their practical programming skills and gaining a greater understanding of the principles.

Everyone interested in computer science, software development, or programming should thoroughly understand data structures and algorithms. It offers a basis for comprehending how software operates, how data is saved and handled, and how to resolve challenging issues. Also, it aids in the development of logical thinking and problem-solving abilities, both of which are necessary for success in any technological profession.

Data structures and algorithms have a plethora of different applications. They are utilised in several different industries, including artificial intelligence, machine learning, cryptography, database administration, networking, and many more. For instance, data structures and algorithms are used in cryptography to securely encrypt and decode data as well as in natural language processing to analyse and comprehend vast volumes of text data.

The purpose of the project work in this course is to provide students hands-on experience using data structures and algorithms to solve real-world problems. Students will gain useful programming abilities and be able to apply the course's concepts to actual challenges.

In conclusion, everyone interested in computer science, software development, or programming should thoroughly understand data structures and algorithms. It offers a basis for comprehending how software operates, how data is saved and handled, and how to resolve challenging issues. Students will gain useful programming abilities and be able to apply course principles to real-world issues thanks to the project work in this course.

**3. Problem Statement**

Design and establish a utopian city where all foundational needs are met, and supplies are scarce yet sufficient. The city should be secure and safe, and it should be capable of offering its population work, a dependable transportation system, access to medical facilities, a better educational system, and industrial services.

**4. Functionalities**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SI. No.** | **Function Name** | **Description** | **DS and Algorithm Used** | **Efficiency** |
|  | floyd(int matrix[20][20], int ); | Shortest path between buildings | Floyd algorithm | O(n^3) |
|  | hashpassword(char []); | Hashing the password of homes | Hashing | 1 |
|  | marathon(); | Traversing through every buildings | BFS | O(v+e) |
|  | search\_house(char [], char []); | Searches the house name to verify the house is present in locality or not | Brute force string Search | O(m\*n) |
|  | flood\_evacuation(); | Connecting the buildings when performing the mock flood evacuation | DFS | O(v+e) |
|  | sort\_industries(); | Sorting industries based on number of employees | Quick sort | O(n\*logn) |
|  | sort\_restaurants(); | Sorting restaurants according to rating | Insertion Sort | O(n\*n) |
|  | kruskals\_sum(struct graph [], int [], int, int ); | Generates minimum spanning tree | Kruskal algorithm | O(elogV) |
|  | TREE \* insert\_into\_bst(TREE \*root,struct college data); | Generates inorder,preorder and postorder traversal of college info | Binary search tree-preorder,inorder,postorder | O(logn) |
|  | Dijkstras(); | Shortest path between other buildings | Dijkstras Algotithm | O(v^2) |

**5. Project Tools**

|  |  |  |
| --- | --- | --- |
| **Si. No.** | **Measure** | **Value** |
|  | Total Functions in Project | 28 |
|  | Total number of lines of code  (Including comments, newlines etc.) | 1983 |
|  | Number of Errors | 0 |
|  | Number of Warnings | 4 |
|  | Team Satisfaction about Project | 100 |

**6. Learning and Takeaway**

What are you taking away as learning from the project? Describe your learning and project takeaways. Also complete the contribution table below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Si. No.** | **Roll** | **Name** | **Contribution**  **Rank** |
|  | 134 | Karthik R Khatavkar | 1 |
|  | 135 | Avaneesh Lad | 1 |
|  | 138 | Apeksha M | 1 |
|  | 139 | Priyanka B M | 1 |

**~\*~\*~\*~\*~\*~\*~\*~**