|  |
| --- |
| **Data Structures and Algorithms** |
| MEN’S BCCI |
| **Course Project Report** |

|  |
| --- |
| **School of Computer Science and Engineering**  **2021-22** |

**Contents**

|  |  |
| --- | --- |
| **Si. No.** | **Topics** |
| 1. | Course and Team Details |
| 2. | Introduction |
| 3. | Problem Definition |
| 4. | Functionalities |
| 5. | Tools and Techniques |
| 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** | B |
| **Year** | 2021-22 |
| **Instructor** | Prakash Hegade |

**1.2 Team Details**

|  |  |  |
| --- | --- | --- |
| **Si. No.** | **Roll No.** | **Name** |
| 1. | 241 | VINAYAK KONE |
| 2. | 243 | SWAROOP |
| 3. | 242 | RAKSHITHA R T |
| 4. | 238 | BHOOMIKA KUMTA |

**2. Introduction**

This course introduces us to data structure and algorithm , goes through the solutions to standard problems in details and give us the insight into how efficient it is to use each one of them. It also teaches the science of evaluating the efficiency of an algorithm.

Data structure helps to store and organize the data and algorithm is a step by step process or formula for problem solving or you can say that it is a set of instruction formulated to conduct a particular work. Its real time application cab be observed through Google map, BCCI, any platform in social media etc. this project strengthens the foundation of problem solving skills, logical thinking as we come across plenty of real time application in our day to day life.

**3. Problem Statement**

We chose the problem statement of replicating an famous app for cricket enthusiast known as BCCI.

Our code performs much functionality seen in app such as

->shows the recent matches, shows player details, upcoming fixtures and many more.

**4. Functionalities**

Here in the time complexity n changes with respect to data stored in files

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SI. No.** | **Function Name** | **Description** | **DS and Algorithm Used** | **TIME COMPLEXITY** |
|  | news\_load | Loads the news details into appropriate data structure | Doubly linked list | O(n) |
|  | News\_Display | Show the latest news headlines | Doubly linked list | O(n) |
|  | Fixtures | Displays the upcoming fixtures of the team | Doubly linked list | Constant time. |
|  | venue\_details | Displays the information about the stadiums in India | Doubly linked list | O(n) |
|  | news\_sort\_by\_dob | Display the news details in the increasing order of their date of publish | Doubly linked list | O(n) |
|  | load\_from\_file | loads the data available into appropriate data structure | file | O(n) |
|  | welcome | prints the welcome message and current IST and UTC date and time | For loop | O(n) |
|  | print\_details | prints the details of all the players | Structure and array | Constant time. |
|  | brute\_force\_search | Here we used the brute force search algorithm to find match for the user given substring. If any match is found we print the details using the below mentioned  Print\_details1 function . | Brute force algorithm. | O(mn) |
|  | print\_details1 | prints the details of the players whose name matches  with the user entered sub string | For loop | O(n) |
|  | sort\_by\_name | Display the player Details in the increasing alphabetical order | Insertion sort algorithm | O(n^2) |
|  | players\_sort\_by\_age | Display the player Details in the increasing order of their age | Bubble sort algorithm | O(n^2) |
|  | players\_sort\_by\_debut\_date | Display the player details from latest debut date to least | Bubble sort algorithm | O(n^2) |
|  | players\_sort\_by\_dob | Display the player details in the increasing order of their dob | Bubble sort algorithm | O(n^2) |
|  | poll\_load | loads the data available into appropriate data structure | file | O(n) |
|  | poll\_store | stores the data of the polls updated by the user back into the file | file | O(n) |
|  | poll | Displays the questions, accepts answers from the user and prints the final result of each question. | For loop. | O(n) |
|  | Facebook | Opens the official Facebook page of the BCCI in browser | System call. | Constant time. |
|  | Instagram | Opens the official Instagram page of the BCCI in browser | System call. | Constant time. |
|  | Twitter | Opens the official twitter page of the BCCI in browser | System call. | Constant time. |
|  | Gallery | Opens the recent photos (Gallery) in external browser | System call. | Constant time. |
|  | Store | Open the official merchandise shopping site in external browser | System call. | Constant time. |
|  | videos | pen the recent videos in external browser | System call. | Constant time. |
|  | IPL | Redirect to the IPL website where all information about IPL is available | System call. | Constant time. |
|  | players\_display\_by\_roll | Display the player details according to role whether batsman/bowler/all-rounder according  to user input | Switch condition. | O(n) |
|  | news | Redirect to the display news function | Double linked list | Constant time |
|  | fixture\_load | Loads the fixture details in the appropriate data structure | files | O(n) |
|  | FIX\_DISPLAY | DISPLAYS THE UPCOMING FIXTURES OF TEAM INDIA | While loop | O(n) |
|  | FIX\_DISPLAY\_BY\_TYPE\_OF\_MATCH | Displays the upcoming match details by specific match type by getting user input | While loop | O(n) |
|  | fix\_display\_by\_month\_range | Displays the upcoming match details between the user given month range for the year 2022 | While loop | O(n) |
| 31. | venue\_details | Loads the Venue details into appropriate data structure and calls the VENUE\_DISPLAY function | File ,while loop | O(n) |
| 32. | VENUE\_DISPLAY | DISPLAYS THE INFORMATION ABOUT THE STADIUMS IN INDIA | While loop | O(n) |
| 33. | news\_sort\_by\_date | Display the news details in the ascending order of their date of publish | Bubble-sort | O(n^2) |
| 34. | results | Gets input from the user about specific match type to display the result | switch | Constant time. |
| 35. | results\_date\_range | Gets input from the user about specific match type to display the result between the date range given by user | switch | Constant time. |
| 36. | results\_odi\_details\_load | Gets the results from the file and load into appropriate structures | file | O(n) |
| 37. | results\_odi\_details | Loads the data into the Doubly Linked list and calls the print function | For loop | O(n) |
| 38. | print\_result\_odi\_t20 | Displays the results of the form ODI and T20I | While loop | O(n) |
| 39. | results\_test\_details\_load | Loads the Test results from file to appropriate data structure | file | O(n) |
| 40. | results\_test\_details | Loads the data into the Doubly Linked list and calls the print function | For loop | O(n) |
| 41. | print\_result\_test | Displays the results of the form TEST | While loop | O(n) |

**5. Tools and Techniques**

**5.1 Data Structures and Algorithms**

We used various data structures and algorithms to implement the project. We used structure array, doubly linked list.

For sorting functions we used-

1. Bubble-sort, Insertion-sort, Heap-sort algorithms.

For sub-string match functions we have used-

1. Brute-force algorithm, Rabin-karp algorithm.

**5.2 Project Statistics**

|  |  |  |
| --- | --- | --- |
| **Si. No.** | **Measure** | **Value** |
|  | Total Functions in Project | 41 |
|  | Total number of lines of code  (Including comments, newlines etc.) | 2570 |
|  | Number of Errors | 0 |
|  | Number of Warnings | 5 |
|  | Team Satisfaction about Project | 95 |

**6. Learning and Takeaway**

This course as well as the project gave different picture and an interesting site how we can change the real time application by using different data structure and algorithm .There might be cases when you will provided with most efficient data structure to work with a robust algorithm. Another critical facet of DSA usage in software development is the time and space constraints. These constraints check the availability and space for an algorithm. An optimized algorithm provides both of these constraints based on availability of resources.

**7. References**

[1] Richard Gilberg, Behrouz A. Forouzan. 2005. Data Structures A Psuedocode Approach with C (2nd ed.). Cengage Learning .

[2] http://people.cs.vt.edu/~shaffer/Book/C++3e20110915.pdf

[3] http://www.cs.toronto.edu/~hojjat/148s07/lectures/week5/07linked.pdf

[4]https://www.quora.com/How-do-I-sort-dates-using-structure-in-the-C-programming-language

[5] https://www.bcci.tv

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