**DSA PROJECT**

***DESCRIPTION:***

**I HAVE TAKEN A FIFA WORLD CUP DATASET FROM 1930 TO 2018. PARSED IT AND IMPLEMENTED DATA STRUCTURE TECHNIQUES TO ENSURE LEAST MEMORY USAGE AND EFFICIENT SEARCHING AND INSERTION OF DATA.**

***STRUCTURES USED:***

|  |  |  |
| --- | --- | --- |
| **#** | **Data Structure** | **Purpose** |
| 1 | **AVL TREES** | **TO STORE DATA HENCE RESULTS IN FASTER SEARCHING.** |
| 2 | **HASH TABLES**  **(ARRAY)** | **GIVES INSTANT ACCESS TO VALUE USING HASH FUNCTION** |
| 3 | **SINGLY LINKED LIST** | **STORE ITEMS NOT USED IN SEARCHING.**  **IT PROVIDES EASY TRAVERSAL.** |

***GOALS:***

1. **Achieving a memory efficient data structure by storing data at one place and using pointers to point to that data when required.**
2. **Use of AVL Trees for better searching functions and better management of data.**
3. **Using hash tables for storing linked list and storing the pointers to the actual data of AVL tress in the nodes of linked list ensuring quick access to the data and less memory consumption.**
4. **Use of linked list for easy traversal of data**
5. **Less focus on creating methods and more focus on data structuring and managing techniques.**

***EXPLANATION:***

* **Match.h declares a class containing all the attributes of a match.**
* **Matchnode.h declares a class for a node containing data from match.h, lchild, rchild.**
* **Matchlist.h declares a class for an avl tree and some functions to insert the matchnode from parser**
* **Into an avl tree. Also, I have made a function that search head to head matches between two teams as an example of how to use this data structure.**
* **Matchtablenode.h declares a class for a node containing two pointers. One point to the objects already stored in the avl trees and one point to the next node.**
* **Matchtablelist.h declares a class for a singly linked list. The matchtablenodes will be stored in the singly linked list. One list will contain nodes representing matches of that specific world cup match.**
* **Worldcupyear.h declares a class for a hash table with data type as matchtablelist as it will store linked list at each array index.**
* **Parser.h which parses the data from the csv file and store it in vector from where I am receiving it as nodes of avl tree. One line per node. All the class objects are created in this class for functioning.**
* **Main.cpp just calls functions from parser1 class and run the functionalities defined in all the included classes.**

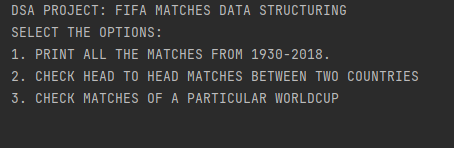
**RUNNING THE PROGRAM:**

**RUNNING THE CODE WILL ASK FOR 3 DIFFERENT OPTIONS FROM THE USER:**

1. **DISPLAY ALL FIFA MATCHES FROM 1930 – 2018.**
2. **FINDING DETAILS OF MATCHES BETWEEN TWO SPECIFIC COUNTRIES.**
3. **FINDING MATCHES OF A PARTICULAR WORLDCUP.**

**(OPTION 1 AND 2 ARE DONE USING SEARCH METHOD DEFINED ON AVL TREES AND OPTION 3 IS DONE USING LINKED LIST STORED IN HASH TABLES. ONCE AGAIN THE FOCUS WAS MORE ON STRUCTURING THE DATA, WE CAN CREATE METHODS EASILY ONCE DATA IS MANAGED)**

**Option when running the code:**

****

**OPTION 1:**

**Calendar

Description automatically generated with medium confidence**

**OPTION 2:**

Text

Description automatically generated

**OPTION 3:**

**Text

Description automatically generated**