

The National Higher School of Artificial Intelligence Data Structures and Algorithms 2

Programming Mini Project: Football Tournament Management System

Submission deadline: Saturday 16/12/2023, not later than 11:59pm

We want to manage a Football tournament, say the Algerian League 1 championship. As you know the tournament has 20 teams, and, as of international competitions requirements, each team has 23 players in addition to the staff. On the field, there are eleven positions, each with a specific role to play. These positions are divided into three main categories: defence, midfield, and attack. Though the player numbers can vary from team to team, they typically are as follows¹:

Number	Football Position
1	Goal Keeper
2	Right Fullback
3	Left Fullback
4	Centre Back
5	Centre Back
6	Defensive Midfielder
7	Right Midfielder
8	Centre Midfielder
9	Striker
10	Attacking Midfielder
11	Left Midfielder

Since there are 23 players on a team and only 11 play on the pitch, each player on the bench will have one of the above positions but obviously with different numbers. The players' numbers are made official for the entire championship and each is obviously unique in a given team.

A team also has an administrative staff and a technical staff. For simplicity, we will consider only the President of the club/team and the coach. The former takes care of the management of the club (administration, finances, recruiting procedures, etc.) while the latter is in charge of training the team and getting as good results as possible.

League games organisation: In a league championship, teams play against each other in a round-robin format, with each team playing every other team twice (once at home and once away). Three (3) points are awarded for a win and one (1) point for a draw, and the team with the most points at the end of the season is declared the winner.

Usually, different statistics are kept. The most common one is the best goal scorer during the entire championship. Depending on the management systems different other stats may be collected and kept.

The football tournament management system (FTMS) we want to build needs to keep all the needed data about the following:

- each game (i.e. the two teams taking part in it and who is the receiving team and the away team), the date and start time of the game, the referee, and the score;
- each player (team and number) who scored a goal for each team;
- each player (team and number) who got a yellow card (warning) and each who got a red card (sent off the pitch);

¹ [Soccer Positions, Numbers, and Roles \(Full Breakdown\) - Soccer Coaching Pro](#) (the spelling has been changed to British spelling for uniformity purposes with the rest of the project statement).

- the duration in minutes of the game. This should not be less than 90 minutes.
- the time (in minutes, starting at zero) each player got onto the pitch (i.e. to play) and the time he was replaced (if he was not replaced, this second time will be the time of the end of the game);
- since the players wear GPS tracking devices during each game, statistics about the numbers of kilometres (a real number) run by each player on the field. (We will neglect the wealth of other information that can be collected about each player.)

This data will be kept and accumulated as appropriate over the years.

Note that a player may be recruited during the year (usually in the winter or summer Mercato) or may have his contract terminated.

The FTMS should organise all this data in such a way that all operations of access to the data and of processing it to return the below results should be done as quickly as possible. (Though you are only asked to simulate the FTMS with data about the League 1 main teams, the FTMS will actually manage all the championships at all levels (regions, etc.) and age categories, male and female teams (which play separate championships).

The FTMS is supposed to be web-based or mobile-based, but this is really optional; you can only implement the strictly necessary functionalities. But nicer systems will be positively appreciated.

Based on EXACTLY² THE AFOREMENTIONED DATA, the results expected from the FTMS are the following, in addition to any other results you may wish (to include and explain in your report why they are good additions).

- A listing of the teams' standings on any date. The standings should show the position of each team from 1st to 20th, the number of games played, number of goals scored, number of goals conceded, the goal difference, and the number of points. Note that, on any day, one may wish to go back in history to any point in time and display these results. For instance, suppose the championship is at week 9, one may want to display the results after week 1 or 5, etc.
- A listing of the goal scorers from the one who scored most goals to the last ones who have scored at least one goal. Here also, one may wish to go back in history to any point in time and display these results. One may wish to draw for any chosen player, the curve of the number of goals he scored against the weeks in the championship.
- A listing for a given team of the kilometres run by each of its players in a given game and the average number of kilometres over all the games played before any point in time (before a selected number of weeks).
- A listing of the best scorers in the competition for a given team as well as among all the teams from the very start of the competition (historically speaking) till the latest week in the competition of the current season.

Part A of the Project:

We have decided to store the data in a Binary Search Tree (plus any other ADTs as you see appropriate)

1. Give the graphical representation of the most suitable ADT for the FTMS. This global ADT could be a combination of ADTs, each of which would be useful for any purposes you will need to explain.
2. Give a complete specification of this ADT.
3. Implement all the operations, making use as much as possible of any ADTs and operations we have studied in class.
4. Write a program which computes the average time for displaying the various lists mentioned above.

² Not less, and not more types of data!

Part B of the Project:

We have decided to store the data in an AVL Tree.

5. Implement all the operations, making use as much as possible of any ADTs and operations we have studied in class.
6. Write a program which computes the average time for displaying the various lists mentioned above.
7. Compare the results obtained in questions 4 and 6, and give your conclusion.

Enjoy!