

League Standing

B. N	Name
1	إبراهيم علي إبراهيم الدسوقي
3	أحمد جمال سعيد
4	أحمد حمدي محمد فهمي
12	آيه عبدالمنعم جابر
21	ريم محمود صالح

GitHub Link:

<https://github.com/algorithm-project-league-standing/league-standing->

Time complexity

Read Data from file (.csv)

```
83  ifstream data;
84  string s, round, date, homeTeam, awayTeam, homeGoals, awayGoals, result;
85  data.open("epl_results.csv", ios::in);
86  getline(data, s, ',');getline(data, s, ',');getline(data, s, ',');getline(data, s, ',');
87  getline(data, s, ',');getline(data, s, ',');getline(data, s, '\n');
88  while(getline(data, round, ',')){
89      getline(data, date, ',');getline(data, homeTeam, ',');getline(data, awayTeam, ',');
90      getline(data, homeGoals, ',');getline(data, awayGoals, ',');getline(data, result, '\n');
91      if(stoint.find(homeTeam) == stoint.end()){
92          nteam++;
93          stoint[homeTeam] = counter++;
94          inttos[counter] = homeTeam;
95      }
96      if(stoint.find(awayTeam) == stoint.end()){
97          nteam++;
98          stoint[awayTeam] = counter++;
99          inttos[counter] = awayTeam;
100     }
101 }
102 data.close();
103
```

$O(E \log V)$



$O(\log V)$



```
119 while(true){
120     vector<Team> teams;
121     for(int i = 0; i <= nteam; ++i) vis[i] = 0;
122     for(auto el : inttos){
123         teams.push_back(Team(el.second));
124     }
125     int trio, choice;
126     cout << nl << "1 - Search by round " << nl << nl << "2 - search by date " << nl << nl << "3 - End The Program " << nl << nl ;
127     cout << "Enter Your Choice : ";
128     cin >> trio ;
129     if(trio == 1){
130         cout << "Enter the round number : ";
131         cin >> choice;
132         for(int i = 0; i < nteam; ++i){
133             if(!vis[i]){
134                 BFS_round(i, choice, teams);
135             }
136         }
137     }
138     else if(trio == 2){
139         cout << "Enter the date in the form 'dd/mm/year' : " ;
140         string da;
141         cin >> da;
142         choice = dateToInt(da);
143         for(int i = 0; i < nteam; ++i){
144             if(!vis[i]){
145                 BFS_date(i, choice, teams);
146             }
147         }
148     }
149     else {
150         break;
151     }
152 }
```

$O(V)$

$O(V)$

$O(V + E)$ using aggregation analysis

```
174 | sort(teams.begin(), teams.end(), srt);
```

$O(V \log V)$ Sorting of teams based on their statistics

```
176 | for(auto el : teams){
177 |     cout<< left << setw(13) << cnt
178 |         << left << setw(23) << el.teamName
179 |         << left << setw(13) << el.matchPlayed
180 |         << left << setw(13) << el.wins
181 |         << left << setw(13) << el.drawns
182 |         << left << setw(13) << el.loses
183 |         << left << setw(13) << el.goalsFor
184 |         << left << setw(13) << el.goalsAgainst
185 |         << left << setw(13) << el.goalsFor - el.goalsAgainst
186 |         << left << setw(13) << el.points
187 |         << nl << nl;
188 |
189 |     out << cnt << ',' << el.teamName << ',' << el.matchPlayed << ',' << el.wins << ',' <<
190 |         el.drawns << ',' << el.loses << ',' << el.goalsFor << ',' << el.goalsAgainst << ',' <<
191 |         el.goalsFor - el.goalsAgainst << ',' << el.points << '\n';
192 |
193 |     cnt++;
194 | }
195 |
196 |
```

$O(V)$

Sorting of teams based on their statistics

```
202 |
203 | int stringToInt(string s){
204 |     int res = 0;
205 |     for(auto el: s)
206 |     {
207 |         res = (res * 10) + (el - '0');
208 |     }
209 |     return res;
210 | }
```

$O(1)$ Function to convert string to int

```
212 | int dateToInt(string s){
213 |     int year, month, day;
214 |     std::sscanf(s.c_str(), "%d/%d/%d", &day, &month, &year) ;
215 |     return 10000 * year + 100 * month + day;
216 | }
```

$O(1)$ Function to convert date to int

```
218 | bool srt(Team t1, Team t2){
219 |     if(t1.points != t2.points) return t1.points > t2.points;
220 |     else if(t1.goalsFor - t1.goalsAgainst != t2.goalsFor - t2.goalsAgainst)
221 |         return t1.goalsFor - t1.goalsAgainst > t2.goalsFor - t2.goalsAgainst;
222 |     else return t1.goalsFor > t2.goalsFor;
223 | }
```

$O(1)$ Compator function

```

224 void BFS_round(int s, int choice, vector<Team> &teams){
225     q.push(s);
226     vis[s] = 1;
227     while(!q.empty()){
228         int u = q.front();
229         q.pop();
230         for(auto v : adj[u]){
231             if(match.round <= choice){
232                 teams[u].goalsFor += match.homeGoals;
233                 teams[v.first].goalsAgainst += match.homeGoals;
234                 teams[u].goalsAgainst += match.awayGoals;
235                 teams[v.first].goalsFor += match.awayGoals;
236                 teams[u].matchPlayed++;
237                 teams[v.first].matchPlayed++;
238                 if(match.result == "H"){
239                     teams[u].wins++;
240                     teams[u].points += 3;
241                     teams[v.first].loses ++;
242                 }
243                 else if(match.result == "A"){
244                     teams[v.first].wins++;
245                     teams[v.first].points += 3;
246                     teams[u].loses ++;
247                 }
248                 else{
249                     teams[u].drawns++;
250                     teams[v.first].drawns++;
251                     teams[u].points++;
252                     teams[v.first].points++;
253                 }
254             }
255             if(!vis[v.first]){
256                 q.push(v.first);
257                 vis[v.first] = 1;
258             }
259         }
260     }
261 }

```

$O(V)$

$O(E + V)$

$O(E)$ using aggregation analysis

The total time complexity :

$$O((V + E) \log V)$$

Space complexity

```
64  int counter, nteam, cnt;
65  map<string, int> stoint;           // O(V)
66  map<int, string> inttos;          // O(V)
67  vector<vector<pair<int, Match>>> adj; // O(V + E)
68  vector<int> vis;                  // O(V)
69  queue<int> q;
70
```