

**Faculty of engineering - Shoubra**

**Benha University**

|  |  |
| --- | --- |
| **Division** | 3rd computer engineering |
| **Academic Year** | 2022-2023 |
| **Course name** | Advanced Algorithm |
| **Course code** | ECE323C |

# Project Title:

**Premier League Table**

**By:**

|  |  |  |
| --- | --- | --- |
| **B.N.** | **Name** | **Email** |
| 15 | بيتر لطيف نصحي | beter195290@feng.bu.edu.eg |

**Approved by:**

|  |  |
| --- | --- |
| Examiners committee | Signature |
| د/ عبدالله سعد |  |
| م/ محمد عبدالمجيد |  |

**Complexity of the code:**

The time complexity of this code is:

* O(E) to build the graph, where E is the number of edges (fixtures)
* O(T) to calculate the results for each team, where N is the number of teams

This is because we memoize the results, so calculating stats for a team is O(1) after the first time.

* O(N log N) to sort the standings, where N is the number of teams.
* O(1) to update the standings for a specific team.

So overall, the time complexity is:

O(E + N + N log N)

Which can be simplified to:

O(E + N log N)

The space complexity remains:

O(N + E)

To store:

* The graph with N nodes and E edges
* The memoized team results
* The standings list

So in summary:

* Time complexity: O(E + N log N)
* Space complexity: O(N + E)

Where:

* E is the number of edges (fixtures)
* N is the number of nodes (teams)

This is an improvement over the original time complexity of O(E\*N) due to memoizing team results and updating standings incrementally.

The constant factors are also improved due to avoiding unnecessary work.