## Complexity Analysis

1. __init__() method: This method initializes two attributes, (num_teams) and (adj_list), which takes  O(V).

2. add_edge() method: This method inserts a new edge into the adjacency list for a given team. Therefore, the time complexity of this method is O(n), where n is the number of matches played by the team up to the current point in time.

3. graph_initialization() method: This method initializes the graph data structure by adding all the edges for each match played in the EPL. The time complexity of this method depends on the number of matches played in the EPL, which is at most equal to the product of the number of teams and the number of rounds played in the league. Therefore, the time complexity of this method is O(n^2), where n is the number of teams in the EPL.

4. calc_standings() method: This method updates the standings of two teams after a match has been played. The method involves updating eight elements of the standings list, which takes constant time. Therefore, the time complexity of this method is O(1).

5. BFS_round() method: This method performs a breadth-first search traversal of the graph data structure to calculate the standings of all the teams up to a given round number. Therefore, the time complexity of this method is O(E+V),

6. BFS_date() method: This method performs a breadth-first search traversal of the graph data structure to calculate the standings of all the teams up to a given date. The time complexity of this method depends on the total number of matches played up to the given date. Therefore, the time complexity of this method is O(EV).