Project Description

Sports Leaderboard Sorting Application

Overview:

This project focuses on sorting a sports leaderboard where each team's statistics are provided in CSV format. Each line contains a team's name along with the number of wins, losses, draws, no result/abandoned matches, and points. The sorting is based on a custom comparator that applies the following rules:

1. **Points (Descending):** Teams with higher points are ranked higher.
2. **Wins (Descending):** If points are the same, the team with more wins ranks higher.
3. **Losses (Ascending):** If wins are the same, the team with fewer losses ranks higher.
4. **Draws (Descending):** If losses are the same, the team with more draws ranks higher.
5. **No Result/Abandoned (Ascending):** If draws are the same, the team with fewer no result/abandoned matches ranks higher.
6. **Team Name (Alphabetical):** As a last resort, teams are sorted alphabetically.

Implementation Details:

- **Input:** A CSV file or list of CSV lines where each line follows the format:
  `TeamName,wins,losses,draws,noResult,points`
- **Processing:**
  - Parse the CSV data.
  - Create an internal representation (e.g., an object or record) for each team.
  - **Modified Sorting:** Instead of sorting the entire list, a heap is built based on the custom comparator. Then, the top k teams are extracted using the heap's extract operation, which takes O(log n) time per extraction, resulting in a total time complexity of O(k log n) for printing the top k teams.
- **Output:** The top k teams from the sorted leaderboard are printed.