Tournament Winner

There’s an algorithms tournament taking place in which teams of programmers compete against each other to solve algorithmic problems as fast as possible. Teams compete in a round robin, where each team faces off against all other teams. Only two teams compete against each other at a time, and for each competition, one team is designated the home team, while the other team is the away team. In each competition there’s always one winner and one loser; there are no ties. A team receives 3 points if it wins and 0 points if it loses. The winner of the tournament is the team that receives the most amount of points.

Given an array of pairs representing the teams that have competed against each other and an array containing the results of each competition, write a function that returns the winner of the tournament. The input arrays are named competitions and results, respectively. The competitions array has elements in the form of [homeTeam, awayTeam], where each team is a string of at most 30 characters representing the name of the team. The results array contains information about the winner of each corresponding competition in the competitions array. Specifically, results[i] denotes the winner of competitions[i], where a 1 in the results array means that the home team in the corresponding competition won and a 0 means that the away team won.

It’s guaranteed that exactly one team will win the tournament and that each team will compete against all other teams exactly once. It’s also guaranteed that the tournament will always have at least two teams.

Sample Input

JS

Copy

```js
competitions = [
  ["HTML", "C#"],
  ["C#", "Python"],
  ["Python", "HTML"],
]
results = [0, 0, 1]
```

Sample Output

JS

Copy

```js
"Python"
// C# beats HTML, Python Beats C#, and Python Beats HTML.
// HTML - 0 points
// C# -  3 points
// Python -  6 points
```

Hints

Hint 1

Don’t overcomplicate this problem. How would you solve it by hand? Consider that approach, and try to translate it into code.

Hint 2

Use a hash table to store the total points collected by each team, with the team names as keys in the hash table. Once you know how many points each team has, how can you determine which one is the winner?

Hint 3

Loop through all of the competitions, and update the hash table at every iteration. For each competition, consider the name of the winning team; if the name already exists in the hash table, update that entry by adding 3 points to it. If the team name doesn’t exist in the hash table, add a new entry in the hash table with the key as the team name and the value as 3 (since the team won its first competition). While looping through all of the competitions, keep track of the team with the highest score, and at the end of the algorithm, return the team with the highest score.

Optimal Space & Time Complexity

O(n) time | O(k) space - where n is the number of competitions and k is the number of teams

JS

Copy

```
/* competitions = [
        ["html", "C#"]
        ["c#", "Python"]
        ["python", "html"]
    ]
    results = [0, 0, 1]

    // loop over competitions
    // make a dict with the name of the team
    // increament the team number which wins
    // return the team with the largest number of wins
*/

function tournamentWinner(competitions, results) {
    const teams = {}
  for (let i = 0; i < competitions.length; i++) {
        if (results[i] === 1) {
            teams[competitions[i][0]] = teams[competitions[i][0]] + 3 || 3
        } else {
            teams[competitions[i][1]] = teams[competitions[i][1]] + 3 || 3
        }
    }
    if (Object.keys(teams).length > 0) {
        let maxValue = Object.keys(teams).reduce((a, b) => teams[a] > teams[b] ? a : b)
        return maxValue
    }
  return [];
}
```

JS

Copy

```
const HOME_TEAM_WON = 1

function tournamentWinner(competitions, results) {
  let currentBestTeam = '';
    const scores = {[currentBestTeam]: 0}

    for (let i = 0; i < competitions.length; i++) {
        const result = results[i];
        let [homeTeam, awayTeam] = competitions[i];

        const winningTeam = result === HOME_TEAM_WON ? homeTeam: awayTeam;

        updateScore(winningTeam, 3, scores)

        if (scores[winningTeam] > scores[currentBestTeam]) {
            currentBestTeam = winningTeam;
        }
    }
  return currentBestTeam;
}

function updateScore(team, points, scores) {
    if (!(team in scores)) scores[team] = 0;
    scores[team] += points;
}

// O(n) time, where n is the number of matches
// O(k) space, where k is the number of teams
```

SHARE ARTICLE Twitter Facebook LinkedIn Mail to Friend

← Minimum Waiting Time

Non-Constructible Change →

Do you have any questions, or simply wish to contact me privately? Don't hesitate to shoot me a DM on Twitter.

Have a wonderful day.

Abhishek 🙏

ABHISHEK KUMAR SINGH

Thanks for reading. It makes a difference. I'll try to help developers with tutorials & blogs. Blogs on design, development and happy-productive life. And it will also support and motivate me to share my knowledge with the community.

PAGES

Articles
Book Notes
About
SiteMap
RSS
Uses
Now
Favourite Links
Algorithms

SOCIAL

Twitter
Instagram
LinkedIn