Description Solution Discuss (333) Submissions

C#

1152. Analyze User Website Visit Pattern

Medium 271 2376 Add to List Share

You are given two string arrays `username` and `website` and an integer array `timestamp`. All the given arrays are of the same length and the tuple `[username[i], website[i], timestamp[i]]` indicates that the user `username[i]` visited the website `website[i]` at time `timestamp[i]`.

A **pattern** is a list of three websites (not necessarily distinct).

- For example, `["home", "away", "love"]`, `["leetcode", "love", "leetcode"]`, and `["luffy", "luffy", "luffy"]` are all patterns.

The **score** of a **pattern** is the number of users that visited all the websites in the pattern in the same order they appeared in the pattern.

- For example, if the pattern is `["home", "away", "love"]`, the score is the number of users `x` such that `x` visited `"home"` then visited `"away"` and visited `"love"` after that.
- Similarly, if the pattern is `["leetcode", "love", "leetcode"]`, the score is the number of users `x` such that `x` visited `"leetcode"` then visited `"love"` and visited `"leetcode"` **one more time** after that.
- Also, if the pattern is `["luffy", "luffy", "luffy"]`, the score is the number of users `x` such that `x` visited `"luffy"` three different times at different timestamps.

Return *the **pattern** with the largest **score***. If there is more than one pattern with the same largest score, return the lexicographically smallest such pattern.

Example 1:

```
Input: username =
["joe","joe","joe","james","james","james","james","mary","mary","mary
 timestamp = [1,2,3,4,5,6,7,8,9,10], website =
["home","about","career","home","cart","maps","home","home","about","c

Output: ["home","about","career"]
Explanation: The tuples in this example are:
["joe","home",1],["joe","about",2],["joe","career",3],
["james","home",4],["james","cart",5],["james","maps",6],
```

```
Dictionary<string,
HashSet<string>>();
        foreach(var kvp
{
            var sites =
kvp.Value;

            for(int i =
sites.Count - 2; i++) {
                for(int
1; j < sites.Count - 1;
                    for
j + 1; k < sites.Count;

= sites[i] + "," + site
"," + sites[k];

 if(counts.ContainsKey(

 counts[key].Add(kvp.Ke


 counts.Add(key, new
HashSet<string>() {kvp.

                    }
                }
            }
        }

        string result =
        int maxPattern
        int currentCoun

        foreach(var kvp
```

Testcase Run Code Result

Accepted Runtime: 136 ms

Your input ["joe","joe","joe [1,2,3,4,5,6,7,8,

Output ["home","about

Expected ["home","about","

Console Use Example Testcase