Description | Editorial | Solutions | Submissions

# 1772. Sort Features by Popularity Premium

Solved

Medium | Topics | Companies | Hint

You are given a string array `features` where `features[i]` is a single word that represents the name of a feature of the latest product you are working on. You have made a survey where users have reported which features they like. You are given a string array `responses`, where each `responses[i]` is a string containing space-separated words.

The **popularity** of a feature is the number of `responses[i]` that contain the feature. You want to sort the features in non-increasing order by their popularity. If two features have the same popularity, order them by their original index in `features`. Notice that one response could contain the same feature multiple times: this feature is only counted once in its popularity.

Return *the features in sorted order.*

**Example 1:**

```
Input: features = ["cooler","lock","touch"], responses = ["i like cooler cooler","lock touch cool","locker like touch"]
Output: ["touch","cooler","lock"]
Explanation: appearances("cooler") = 1, appearances("lock") = 1, appearances("touch") = 2. Since "cooler" and "lock" both had 1 appearance, "cooler" comes first because "cooler" came first in the features array.
```

**Example 2:**

```
Input: features = ["a","aa","b","c"], responses = ["a","a aa","a a a a a","b a"]
Output: ["a","aa","b","c"]
```

**Constraints:**

- `1 <= features.length <= 10^4`
- `1 <= features[i].length <= 10`
- `features` contains no duplicates.
- `features[i]` consists of lowercase letters.
- `1 <= responses.length <= 10^2`
- `1 <= responses[i].length <= 10^3`
- `responses[i]` consists of lowercase letters and spaces.
- `responses[i]` contains no two consecutive spaces.
- `responses[i]` has no leading or trailing spaces.

Seen this question in a real interview before? 1/5

Yes | No

Accepted 5.9K | Submissions 9.1K | Acceptance Rate 65.6%

Topics

Companies

Hint 1

Hint 2

Similar Questions

Discussion (1)



87 | 1

Code

Python3 • Auto

```python
class Solution:
    def sortFeatures(self, features: List[str], responses: List[str]) -> List[str]:

        dictx = {}

        for each in features:
            dictx[each] = 0

        for each in responses:
            for val in set(list(map(str, each.split(" ")))):
                if val in dictx:
                    dictx[val] += 1

        dictx = sorted(dictx.items(), key = lambda x: x[1], reverse=True)

        ans = []

        for each in dictx:
            ans.append(each[0])
```

Saved | Ln 13, Col 1

Testcase | Test Result

Accepted Runtime: 28 ms

Case 1 • Case 2

Input

features =
["cooler","lock","touch"]

responses =
["i like cooler cooler","lock touch cool","locker like touch"]

Output

["touch","cooler","lock"]