(/problems ১০০ rain-water

## 6183. Sum of Prefix Scores of Strings

My Submissions (/contest/weekly-contest-311/problems/sum-of-prefix-scores-of-strings/submissions/)

Back to Contest (/contest/weekly-contest-311/)

You are given an array words of size n consisting of non-empty strings.

We define the **score** of a string word as the **number** of strings words [i] such that word is a **prefix** of words [i] .

• For example, if words = ["a", "ab", "abc", "cab"], then the score of "ab" is 2, since "ab" is a prefix of both "ab" and "abc".

Return an array answer of size n where answer[i] is the sum of scores of every non-empty prefix of words[i].

Note that a string is considered as a prefix of itself.

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Hard

## Example 1:

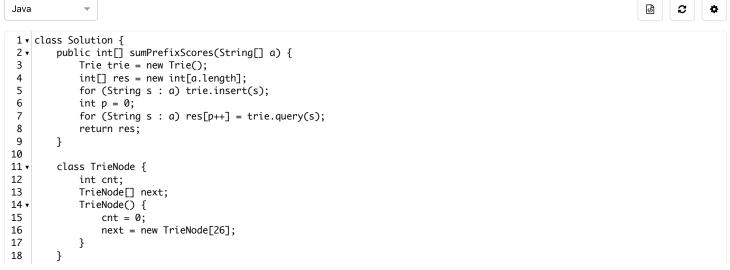
```
Input: words = ["abc","ab","bc","b"]
Output: [5,4,3,2]
Explanation: The answer for each string is the following:
        "abc" has 3 prefixes: "a", "ab", and "abc".
        There are 2 strings with the prefix "a", 2 strings with the prefix "ab", and 1 string with the prefix "abc".
The total is answer[0] = 2 + 2 + 1 = 5.
        "ab" has 2 prefixes: "a" and "ab".
        There are 2 strings with the prefix "a", and 2 strings with the prefix "ab".
The total is answer[1] = 2 + 2 = 4.
        "bc" has 2 prefixes: "b" and "bc".
        There are 2 strings with the prefix "b", and 1 string with the prefix "bc".
The total is answer[2] = 2 + 1 = 3.
        "b" has 1 prefix: "b".
        There are 2 strings with the prefix "b".
The total is answer[3] = 2.
```

## Example 2:

```
Input: words = ["abcd"]
Output: [4]
Explanation:
"abcd" has 4 prefixes: "a", "ab", "abc", and "abcd".
Each prefix has a score of one, so the total is answer[0] = 1 + 1 + 1 + 1 = 4.
```

## **Constraints:**

- 1 <= words.length <= 1000
- 1 <= words[i].length <= 1000
- words[i] consists of lowercase English letters.



```
19
20 ▼
        class Trie {
            TrieNode root;
21
22 🔻
            Trie() {
23
                 root = new TrieNode();
24
25 ▼
             void insert(String s) {
                 TrieNode cur = root;
26
27 ▼
                 for (int i = 0; i < s.length(); i++) {
                     int idx = s.charAt(i) - 'a';
28
                     if (cur.next[idx] == null) cur.next[idx] = new TrieNode();
29
30
                     cur = cur.next[idx];
31
                     cur.cnt++;
32
                 }
33
34 ▼
             int query(String s) {
35
                 TrieNode cur = root;
                 int res = 0;
36
                 for (int i = 0; i < s.length(); i++) {
37 ▼
38
                     int idx = s.charAt(i) - 'a';
39
                     cur = cur.next[idx];
40
                     res += cur.cnt;
41
                 }
42
                 return res;
43
            }
44
        }
45
    }
```

 $\ \square$  Custom Testcase

Use Example Testcases

Submission Result: Accepted (/submissions/detail/802644352/) 2

More Details > (/submissions/detail/802644352/)

Run

**△** Submit

Share your acceptance!

**⋖**3

Copyright © 2022 LeetCode

Help Center (/support) | Jobs (/jobs) | Bug Bounty (/bugbounty) | Online Interview (/interview/) | Students (/student) | Terms (/terms) | Privacy Policy (/privacy)

United States (/region)