

Problem List

AcceptedEditorialSubmissionsSolutions

245. Shortest Word Distance III

Premium

Solved

Medium

Topics

Companies

Given an array of strings `wordsDict` and two strings that already exist in the array `word1` and `word2`, return *the shortest distance between the occurrence of these two words in the list*.

Note that `word1` and `word2` may be the same. It is guaranteed that they represent **two individual words** in the list.

Example 1:

Input: `wordsDict = ["practice", "makes", "perfect", "coding", "makes"], word1 = "makes", word2 = "coding"`

Output: 1

Example 2:

Input: `wordsDict = ["practice", "makes", "perfect", "coding", "makes"], word1 = "makes", word2 = "makes"`

Output: 3

Constraints:

- `1 <= wordsDict.length <= 105`
- `1 <= wordsDict[i].length <= 10`
- `wordsDict[i]` consists of lowercase English letters.
- `word1` and `word2` are in `wordsDict`.

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

Yes

No

Accepted 82.1K

Submissions 140.4K

Acceptance Rate 58.5%

Topics

Companies

Similar Questions

Discussion (4)

Copyright © 2024 LeetCode All rights reserved

503

4

</> Code

Python3

Auto

```
1 class Solution:
2     def shortestWordDistance(self, wordsDict: List[str], word1: str, word2: str) -> int:
3
4         dictx = {}
5
6         for i in range(0, len(wordsDict)):
7             if wordsDict[i] not in dictx:
8                 dictx[wordsDict[i]] = [i]
9             else:
10                 dictx[wordsDict[i]].append(i)
11
12         min_distnce = math.inf
13
14         if word1 == word2:
15             listx = dictx[word1]
16
17             for i in range(0, len(listx)):
18                 for j in range(i+1, len(listx)):
19                     val = abs(listx[i]-listx[j])
20                     if val < min_distnce:
21                         min_distnce = val
22                     if min_distnce == 1:
23                         return 1
24
25         else:
26             for i in dictx[word1][::-1]:
27                 for j in dictx[word2][::-1]:
28                     val = abs(i-j)
29                     if val < min_distnce:
30                         min_distnce = val
31                     if min_distnce == 1:
32                         return 1
33
34         return min_distnce
```

Saved

Ln 17, Col 43

Testcase

Test Result

word1 =

"coding"

Stdout

1

4

3