

2452. Words Within Two Edits of Dictionary

My Submissions (/contest/biweekly-contest-90/problems/words-within-two-edits-of-dictionary/submissions/)

Back to Contest (/contest/biweekly-contest-90/)

You are given two string arrays, `queries` and `dictionary`. All words in each array comprise of lowercase English letters and have the same length.

In one **edit** you can take a word from `queries`, and change any letter in it to any other letter. Find all words from `queries` that, after a **maximum** of two edits, equal some word from `dictionary`.

Return a list of all words from `queries`, that match with some word from `dictionary` after a maximum of **two edits**. Return the words in the **same order** they appear in `queries`.

User Accepted:	6969
User Tried:	7443
Total Accepted:	7134
Total Submissions:	12564
Difficulty:	Medium

Example 1:

**Input:** `queries = ["word","note","ants","wood"], dictionary = ["wood","joke","moat"]`  
**Output:** `["word","note","wood"]`  
**Explanation:**

- Changing the 'r' in "word" to 'o' allows it to equal the dictionary word "wood".
- Changing the 'n' to 'j' and the 't' to 'k' in "note" changes it to "joke".
- It would take more than 2 edits for "ants" to equal a dictionary word.
- "wood" can remain unchanged (0 edits) and match the corresponding dictionary word.

Thus, we return `["word","note","wood"]`.

Example 2:

**Input:** `queries = ["yes"], dictionary = ["not"]`  
**Output:** `[]`  
**Explanation:**

Applying any two edits to "yes" cannot make it equal to "not". Thus, we return an empty array.

Constraints:

- `1 <= queries.length, dictionary.length <= 100`
- `n == queries[i].length == dictionary[j].length`
- `1 <= n <= 100`
- All `queries[i]` and `dictionary[j]` are composed of lowercase English letters.

Discuss (https://leetcode.com/problems/words-within-two-edits-of-dictionary/discuss)

JavaScript

📄 ↺ ⚙️

```
1 const twoEditWords = (a, b) => {
2   let res = [];
3   for (const s of a) {
4     let ok = false;
5     for (const t of b) {
6       if (can(s, t)) {
7         ok = true;
8         break;
9       }
10    }
11    if (ok) res.push(s);
12  }
13  return res;
14 };
15
16 const can = (s, t) => {
17   let n = s.length, diff = 0;
18   for (let i = 0; i < n; i++) {
19     if (s[i] !== t[i]) diff++;
20   }
21   return diff <= 2;
22 };
```

☐ Custom Testcase

Use Example Testcases

Run

Submit

Submission Result: **Accepted** (/submissions/detail/833763492/) ?

More Details > (/submissions/detail/833763492/)

Share your acceptance!

3