5302. Encrypt and Decrypt Strings

My Submissions (/contest/weekly-contest-287/problems/encrypt-and-decrypt-strings/submissions/)

Back to Contest (/contest/weekly-contest-287/)

ref=nb_npl)

You are given a character array keys containing **unique** characters and a string array values containing strings of length 2. You are also given another string array dictionary that contains all permitted original strings after decryption. You should implement a data structure that can encrypt or decrypt a **0-indexed** string.

A string is **encrypted** with the following process:

- 1. For each character c in the string, we find the index i satisfying keys [i] == c in keys.
- 2. Replace c with values[i] in the string.

A string is **decrypted** with the following process:

- User Accepted: 0

 User Tried: 0

 Total Accepted: 0

 Total Submissions: 0

 Difficulty: (Hard)
- 1. For each substring s of length 2 occurring at an even index in the string, we find an i such that values[i] == s. If there are multiple valid i, we choose **any** one of them. This means a string could have multiple possible strings it can decrypt to.
- 2. Replace s with keys[i] in the string.

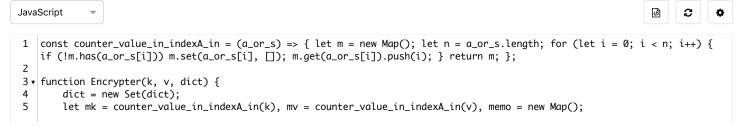
Implement the Encrypter class:

- Encrypter(char[] keys, String[] values, String[] dictionary) Initializes the Encrypter class with keys, values, and dictionary.
- String encrypt(String word1) Encrypts word1 with the encryption process described above and returns the encrypted string.
- int decrypt(String word2) Returns the number of possible strings word2 could decrypt to that also appear in dictionary.

Example 1:

Constraints:

- 1 <= keys.length == values.length <= 26
- values[i].length == 2
- 1 <= dictionary.length <= 100
- 1 <= dictionary[i].length <= 100
- All keys[i] and dictionary[i] are unique.
- 1 <= word1.length <= 2000
- 1 <= word2.length <= 200
- All word1[i] appear in keys.
- word2.length is even.
- keys, values[i], dictionary[i], word1, and word2 only contain lowercase English letters.
- At most 200 calls will be made to encrypt and decrypt in total.



```
return { encrypt, decrypt }
 6
 7 ▼
        function encrypt(s) {
            let res = '';
 8
             for (const c of s) {
 9 ▼
10
                let i = mk.get(c)[0];
                res += v[i];
11
12
13
            return res;
14
        function decrypt(s) {
15 ▼
16
            if (memo.has(s)) return memo.get(s);
             let res = 0;
17
18 ▼
            for (const dic of dict) {
19
                if (encrypt(dic) == s) res++;
20
21
            memo.set(s, res);
22
             return res;
23
        }
24
    }
```

 $\ \square$ Custom Testcase

Use Example Testcases

Share your acceptance!

Copyright © 2022 LeetCode

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

United States (/region)

Run

△ Submit