

Problem 2531: Make Number of Distinct Characters Equal

Problem Information

Difficulty: Medium

Acceptance Rate: 27.33%

Paid Only: No

Tags: Hash Table, String, Counting

Problem Description

You are given two **0-indexed** strings `word1` and `word2`.

A **move** consists of choosing two indices `i` and `j` such that `0 ≤ i < word1.length` and `0 ≤ j < word2.length` and swapping `word1[i]` with `word2[j]`.

Return `true` if it is possible to get the number of distinct characters in `word1` and `word2` to be equal with **exactly one** move. Return `false` otherwise.

Example 1:

Input: `word1 = "ac", word2 = "b"` **Output:** `false` **Explanation:** Any pair of swaps would yield two distinct characters in the first string, and one in the second string.

Example 2:

Input: `word1 = "abcc", word2 = "aab"` **Output:** `true` **Explanation:** We swap index 2 of the first string with index 0 of the second string. The resulting strings are `word1 = "abac"` and `word2 = "cab"`, which both have 3 distinct characters.

Example 3:

Input: `word1 = "abcde", word2 = "fghij"` **Output:** `true` **Explanation:** Both resulting strings will have 5 distinct characters, regardless of which indices we swap.

****Constraints:****

* `1 <= word1.length, word2.length <= 105` * `word1` and `word2` consist of only lowercase English letters.

Code Snippets

C++:

```
class Solution {
public:
    bool isItPossible(string word1, string word2) {

    }
};
```

Java:

```
class Solution {
    public boolean isItPossible(String word1, String word2) {

    }
}
```

Python3:

```
class Solution:
    def isItPossible(self, word1: str, word2: str) -> bool:
```