Problem 1657: Determine if Two Strings Are Close

Problem Information

Difficulty: Medium
Acceptance Rate: 54.21%
Paid Only: No
Tags: Hash Table, String, Sorting, Counting

Problem Description

Two strings are considered **close** if you can attain one from the other using the following operations:

* Operation 1: Swap any two **existing** characters. * For example, `a _b_ cd _e_ -> a _e_ cd _b_` * Operation 2: Transform **every** occurrence of one **existing** character into another **existing** character, and do the same with the other character. * For example, `_aa_ c _abb_ -> _bb_ c _baa_` (all `a`'s turn into `b`'s, and all `b`'s turn into `a`'s)

You can use the operations on either string as many times as necessary.

Given two strings, `word1` and `word2`, return `true` _if_`word1` _and_`word2` _are**close**_ , and _`false` _otherwise._

Example 1:

Input: word1 = "abc", word2 = "bca" **Output:** true **Explanation:** You can attain word2 from word1 in 2 operations. Apply Operation 1: "a _bc_ " -> "a _cb_ " Apply Operation 1: "_a_ c _b_ " -> "_b_ c _a_ "

Example 2:

Input: word1 = "a", word2 = "aa" **Output:** false **Explanation:** It is impossible to attain word2 from word1, or vice versa, in any number of operations.

Example 3:

Input: word1 = "cabbba", word2 = "abbccc" **Output:** true **Explanation:** You can attain word2 from word1 in 3 operations. Apply Operation 1: "ca _b_ bb _a_ " -> "ca _a_ bb _b_ " Apply Operation 2: "_c_ aa _bbb_ " -> "_b_ aa _ccc_ " Apply Operation 2: "_baa_ ccc" -> "_abb_ ccc"

Constraints:

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

Code Snippets

C++:

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


}
};
```

Java:

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


}
}
```

Python3:

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