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, abcde -> aecdb
* Operation 2: Transform **every** occurrence of one **existing** character into another **existing** character, and do the same with the other character.
  + For example, aacabb -> bbcbaa (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: "abc" -> "acb"  
Apply Operation 1: "acb" -> "bca"

**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: "cabbba" -> "caabbb"  
Apply Operation 2: "caabbb" -> "baaccc"  
Apply Operation 2: "baaccc" -> "abbccc"

**Constraints:**

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