Problem 854: K-Similar Strings

Problem Information

Difficulty: Hard
Acceptance Rate: 40.40%
Paid Only: No
Tags: Hash Table, String, Breadth-First Search

Problem Description

Strings `s1` and `s2` are `k`**-similar** (for some non-negative integer `k`) if we can swap the positions of two letters in `s1` exactly `k` times so that the resulting string equals `s2`.

Given two anagrams `s1` and `s2`, return the smallest `k` for which `s1` and `s2` are `k`**-similar**.

Example 1:

Input: s1 = "ab", s2 = "ba" **Output:** 1 **Explanation:** The two string are 1-similar because we can use one swap to change s1 to s2: "ab" --> "ba".

Example 2:

Input: s1 = "abc", s2 = "bca" **Output:** 2 **Explanation:** The two strings are 2-similar because we can use two swaps to change s1 to s2: "abc" --> "bac" --> "bca".

Constraints:

* `1 <= s1.length <= 20` * `s2.length == s1.length` * `s1` and `s2` contain only lowercase letters from the set `{'a', 'b', 'c', 'd', 'e', 'f'}`. * `s2` is an anagram of `s1`.

Code Snippets

C++:

```
class Solution {
public:
int kSimilarity(string s1, string s2) {

}
};
```

Java:

```
class Solution {
public int kSimilarity(String s1, String s2) {

}
}
```

Python3:

```
class Solution:
def kSimilarity(self, s1: str, s2: str) -> int:
```