Problem 1247: Minimum Swaps to Make Strings Equal

Problem Information

Difficulty: Medium
Acceptance Rate: 65.06%
Paid Only: No
Tags: Math, String, Greedy

Problem Description

You are given two strings `s1` and `s2` of equal length consisting of letters `"x"` and `"y"` **only**. Your task is to make these two strings equal to each other. You can swap any two characters that belong to **different** strings, which means: swap `s1[i]` and `s2[j]`.

Return the minimum number of swaps required to make `s1` and `s2` equal, or return `-1` if it is impossible to do so.

Example 1:

Input: s1 = "xx", s2 = "yy" **Output:** 1 **Explanation:** Swap s1[0] and s2[1], s1 = "yx", s2 = "yx".

Example 2:

Input: s1 = "xy", s2 = "yx" **Output:** 2 **Explanation:** Swap s1[0] and s2[0], s1 = "yy", s2 = "xx". Swap s1[0] and s2[1], s1 = "xy", s2 = "xy". Note that you cannot swap s1[0] and s1[1] to make s1 equal to "yx", cause we can only swap chars in different strings.

Example 3:

Input: s1 = "xx", s2 = "xy" **Output:** -1

Constraints:

* `1 <= s1.length, s2.length <= 1000` * `s1.length == s2.length` * `s1, s2` only contain `'x'` or `'y'`.

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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