Problem 1864: Minimum Number of Swaps to Make the Binary String Alternating

Problem Information

Difficulty: Medium
Acceptance Rate: 43.70%
Paid Only: No
Tags: String, Greedy

Problem Description

Given a binary string `s`, return _the**minimum** number of character swaps to make it **alternating** , or _`-1` _if it is impossible._

The string is called **alternating** if no two adjacent characters are equal. For example, the strings `"010"` and `"1010"` are alternating, while the string `"0100"` is not.

Any two characters may be swapped, even if they are **not adjacent**.

Example 1:

Input: s = "111000" **Output:** 1 **Explanation:** Swap positions 1 and 4: "1 _1_ 10 _0_ 0" -> "1 _0_ 10 _1_ 0" The string is now alternating.

Example 2:

Input: s = "010" **Output:** 0 **Explanation:** The string is already alternating, no swaps are needed.

Example 3:

Input: s = "1110" **Output:** -1

Constraints:

* `1 <= s.length <= 1000` * `s[i]` is either `'0'` or `'1'`.

Code Snippets

C++:

```
class Solution {
public:
int minSwaps(string s) {

}
};
```

Java:

```
class Solution {
public int minSwaps(String s) {

}
}
```

Python3:

```
class Solution:
def minSwaps(self, s: str) -> int:
```