Problem 1758: Minimum Changes To Make Alternating Binary String

Problem Information

Difficulty: Easy
Acceptance Rate: 63.74%
Paid Only: No
Tags: String

Problem Description

You are given a string `s` consisting only of the characters `'0'` and `'1'`. In one operation, you can change any `'0'` to `'1'` or vice versa.

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

Return _the**minimum** number of operations needed to make_ `s` _alternating_.

Example 1:

Input: s = "0100" **Output:** 1 **Explanation:** If you change the last character to '1', s will be "0101", which is alternating.

Example 2:

Input: s = "10" **Output:** 0 **Explanation:** s is already alternating.

Example 3:

Input: s = "1111" **Output:** 2 **Explanation:** You need two operations to reach "0101" or "1010".

Constraints:

* `1 <= s.length <= 104` * `s[i]` is either `'0'` or `'1'`.

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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