Problem 1888: Minimum Number of Flips to Make the Binary String Alternating

Problem Information

Difficulty: Medium
Acceptance Rate: 40.84%
Paid Only: No
Tags: String, Dynamic Programming, Sliding Window

Problem Description

You are given a binary string `s`. You are allowed to perform two types of operations on the string in any sequence:

* **Type-1: Remove** the character at the start of the string `s` and **append** it to the end of the string. * **Type-2: Pick** any character in `s` and **flip** its value, i.e., if its value is `'0'` it becomes `'1'` and vice-versa.

Return _the**minimum** number of **type-2** operations you need to perform_ _such that_`s` _becomes**alternating**._

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.

Example 1:

Input: s = "111000" **Output:** 2 **Explanation** : Use the first operation two times to make s = "100011". Then, use the second operation on the third and sixth elements to make s = "10 _1_ 01 _0_ ".

Example 2:

Input: s = "010" **Output:** 0 **Explanation** : The string is already alternating.

Example 3:

Input: s = "1110" **Output:** 1 **Explanation** : Use the second operation on the second element to make s = "1 _0_ 10".

Constraints:

* `1 <= s.length <= 105` * `s[i]` is either `'0'` or `'1'`.

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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