Problem 2380: Time Needed to Rearrange a Binary String

Problem Information

Difficulty: Medium
Acceptance Rate: 52.06%
Paid Only: No
Tags: String, Dynamic Programming, Simulation

Problem Description

You are given a binary string `s`. In one second, **all** occurrences of `"01"` are **simultaneously** replaced with `"10"`. This process **repeats** until no occurrences of `"01"` exist.

Return _the number of seconds needed to complete this process._

Example 1:

Input: s = "0110101" **Output:** 4 **Explanation:** After one second, s becomes "1011010". After another second, s becomes "1101100". After the third second, s becomes "1110100". After the fourth second, s becomes "1111000". No occurrence of "01" exists any longer, and the process needed 4 seconds to complete, so we return 4.

Example 2:

Input: s = "11100" **Output:** 0 **Explanation:** No occurrence of "01" exists in s, and the processes needed 0 seconds to complete, so we return 0.

Constraints:

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

Follow up:

Can you solve this problem in O(n) time complexity?

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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