

Problem 926: Flip String to Monotone Increasing

Problem Information

Difficulty: Medium

Acceptance Rate: 61.74%

Paid Only: No

Tags: String, Dynamic Programming

Problem Description

A binary string is monotone increasing if it consists of some number of `0`'s (possibly none), followed by some number of `1`'s (also possibly none).

You are given a binary string `s`. You can flip `s[i]` changing it from `0` to `1` or from `1` to `0`.

Return `the minimum number of flips to make `s` monotone increasing`.

Example 1:

Input: `s = "00110"` **Output:** `1` **Explanation:** We flip the last digit to get 00111.

Example 2:

Input: `s = "010110"` **Output:** `2` **Explanation:** We flip to get 011111, or alternatively 000111.

Example 3:

Input: `s = "00011000"` **Output:** `2` **Explanation:** We flip to get 00000000.

Constraints:

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

Code Snippets

C++:

```
class Solution {  
public:  
    int minFlipsMonoIncr(string s) {  
  
    }  
};
```

Java:

```
class Solution {  
    public int minFlipsMonoIncr(String s) {  
  
    }  
}
```

Python3:

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