Problem 3228: Maximum Number of Operations to Move Ones to the End

Problem Information

Difficulty: Medium
Acceptance Rate: 67.04%
Paid Only: No
Tags: String, Greedy, Counting

Problem Description

You are given a binary string `s`.

You can perform the following operation on the string **any** number of times:

* Choose **any** index `i` from the string where `i + 1 < s.length` such that `s[i] == '1'` and `s[i + 1] == '0'`. * Move the character `s[i]` to the **right** until it reaches the end of the string or another `'1'`. For example, for `s = "010010"`, if we choose `i = 1`, the resulting string will be `s = "0** _001_** 10"`.

Return the **maximum** number of operations that you can perform.

Example 1:

Input: s = "1001101"

Output: 4

Explanation:

We can perform the following operations:

* Choose index `i = 0`. The resulting string is `s = "_**001**_ 1101"`. * Choose index `i = 4`. The resulting string is `s = "0011 _**01**_ 1"`. * Choose index `i = 3`. The resulting string is `s = "001** _01_** 11"`. * Choose index `i = 2`. The resulting string is `s = "00** _01_** 111"`.

Example 2:

Input: s = "00111"

Output: 0

Constraints:

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

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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