Problem 1221: Split a String in Balanced Strings

Problem Information

Difficulty: Easy
Acceptance Rate: 87.11%
Paid Only: No
Tags: String, Greedy, Counting

Problem Description

Balanced strings are those that have an equal quantity of `'L'` and `'R'` characters.

Given a **balanced** string `s`, split it into some number of substrings such that:

* Each substring is balanced.

Return _the**maximum** number of balanced strings you can obtain._

Example 1:

Input: s = "RLRRLLRLRL" **Output:** 4 **Explanation:** s can be split into "RL", "RRLL", "RL", "RL", each substring contains same number of 'L' and 'R'.

Example 2:

Input: s = "RLRRRLLRLL" **Output:** 2 **Explanation:** s can be split into "RL", "RRRLLRLL", each substring contains same number of 'L' and 'R'. Note that s cannot be split into "RL", "RR", "RL", "LR", "LL", because the 2nd and 5th substrings are not balanced.

Example 3:

Input: s = "LLLLRRRR" **Output:** 1 **Explanation:** s can be split into "LLLLRRRR".

Constraints:

* `2 <= s.length <= 1000` * `s[i]` is either `'L'` or `'R'`. * `s` is a **balanced** string.

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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