

1221. 分割平衡字符串

难度：简单

标签：贪心算法，字符串

在一个平衡字符串中，'L' 和 'R' 字符的数量是相同的。

给你一个平衡字符串 *s*，请你将它分割成尽可能多的平衡字符串。

注意：分割得到的每个字符串都必须是平衡字符串。

返回可以通过分割得到的平衡字符串的最大数量。

示例 1：

输入：s = "RLRRLLRLRL" 输出：4 解释：s 可以分割为 "RL"、"RRL"、"RL"、"RL"，每个子字符串中都包含相同数量的 'L' 和 'R'。示例 2：

输入：s = "RLLLLRRRLR" 输出：3 解释：s 可以分割为 "RL"、"LLLR"、"RL"，每个子字符串中都包含相同数量的 'L' 和 'R'。示例 3：

输入：s = "LLLLRRRR" 输出：1 解释：s 只能保持原样 "LLLLRRRR"。示例 4：

输入：s = "RLRRRLRLRL" 输出：2 解释：s 可以分割为 "RL"、"RRRLRL"，每个子字符串中都包含相同数量的 'L' 和 'R'。

提示：

1 <= s.length <= 1000 s[i] = 'L' 或 'R' s 是一个平衡字符串

```
class Solution:
    def balancedStringSplit(self, s: str) -> int:
        numSL = 0 # L字符的次数
        numSR = 0 # R字符的次数
        ans = 0
        for i in s:
            if i == 'L':
                numSL += 1
            else:
                numSR += 1
            if numSL == numSR:
                ans += 1
        return ans
```

时间复杂度O(n)，空间复杂度O(1)

优化

```
class Solution:
    def balancedStringSplit(self, s: str) -> int:
        nums = 0
        ans = 0
        for i in s:
            if i == 'L':
                nums += 1
            else:
                nums -= 1
            if nums == 0:
                ans += 1
        return ans
```