# 题目

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子字符串，并满足：

每个子字符串都是平衡字符串。

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"、"RRLL"、"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 = "RLRRRLLRLL"

输出：2

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

注意，s 无法分割为 "RL"、"RR"、"RL"、"LR"、"LL" 因为第 2 个和第 5 个子字符串不是平衡字符串。

示例 3：

输入：s = "LLLLRRRR"

输出：1

解释：s 只能保持原样 "LLLLRRRR" 。

提示：

2 <= s.length <= 1000

s[i] = 'L' 或 'R'

s 是一个 平衡 字符串

# 分析

## 方法一：贪心

class Solution {

public:

int balancedStringSplit(string s) {

int ans = 0, d = 0;

for (char ch : s) {

ch == 'L' ? ++d : --d;

if (d == 0) {

++ans;

}

}

return ans;

}

};

复杂度分析：

时间复杂度：O(n)，其中n是字符串s的长度。我们仅需遍历s一次。

空间复杂度：O(1)。只需要常数的空间存放若干变量。