# 题目

如果一个字符串不含有任何重复字符，我们称这个字符串为好字符串。

给你一个字符串s，请你返回s中长度为3的好子字符串的数量。

注意，如果相同的好子字符串出现多次，每一次都应该被记入答案之中。

子字符串是一个字符串中连续的字符序列。

示例 1：

输入：s = "xyzzaz"

输出：1

解释：总共有 4 个长度为 3 的子字符串："xyz"，"yzz"，"zza" 和 "zaz" 。

唯一的长度为 3 的好子字符串是 "xyz" 。

示例 2：

输入：s = "aababcabc"

输出：4

解释：总共有 7 个长度为 3 的子字符串："aab"，"aba"，"bab"，"abc"，"bca"，"cab" 和 "abc" 。

好子字符串包括 "abc"，"bca"，"cab" 和 "abc" 。

提示：

1 <= s.length <= 100

s只包含小写英文字母。

# 分析

## 方法一：遍历起始下标

思路：

我们遍历子字符串的起始下标i，并检验i开头的长度为3的子串是否为好字符串，即是否不含有重复字符。与此同时，我们维护长度为3好子串的个数。

代码：

class Solution {

public:

int countGoodSubstrings(string s) {

int res = 0;

int n = s.size();

for (int i = 0; i < n - 2; i++) {

if (s[i] != s[i+1] && s[i] != s[i+2] &&

s[i+1] != s[i+2]) {

++res;

}

}

return res;

}

};

复杂度分析：

时间复杂度：O(n)，其中n为s的长度。我们遍历了一遍字符串，对于每个下标为i且长度为3的子字符串，判断其是不是好子字符串的时间复杂度为O(1)。

空间复杂度：O(1)。