

5: Leetcode 1876. Substrings of Size Three with Distinct Characters

⇒ <https://leetcode.com/problems/substrings-of-size-three-with-distinct-characters/>

Description

The screenshot shows the LeetCode interface for problem 1876. At the top, there's a navigation bar with 'Problem List', 'Run', and 'S' buttons. Below that, tabs for 'Description', 'Accepted', 'Editorial', 'Solutions', and 'Submissions' are visible. The problem title '1876. Substrings of Size Three with Distinct Characters' is prominently displayed, followed by 'Solved' with a green checkmark. Below the title, there are buttons for 'Easy', 'Topics', 'Companies', and 'Hint'. The problem description states: 'A string is **good** if there are no repeated characters. Given a string `s`, return the number of **good substrings** of length **three** in `s`. Note that if there are multiple occurrences of the same substring, every occurrence should be counted. A **substring** is a contiguous sequence of characters in a string.'

Example 1:

Input: `s = "xyzzaz"`
Output: 1
Explanation: There are 4 substrings of size 3: "xyz", "yzz", "zza", and "zaz". The only good substring of length 3 is "xyz".

Example 2:

Input: `s = "aababcabc"`
Output: 4
Explanation: There are 7 substrings of size 3: "aab", "aba", "bab", "abc", "bca", "cab", and "abc". The good substrings are "abc", "bca", "cab", and "abc".

At the bottom, there are social interaction buttons: '1.5K' likes, '13' comments, and icons for share, star, and help.

2 - Code-

better

```
class Solution:
    def countGoodSubstrings(self, s: str) -> int:
        count=0
        for i in range(len(s)-2):
```

```
if(len(set(s[i:i+3])) == 3):  
    count +=1  
return count
```

```
class Solution:  
    def countGoodSubstrings(self, s: str) -> int:  
        count = 0  
        for i in range(2, len(s)):  
            if s[i] != s[i-1] and s[i] != s[i-2] and s[i-1] != s[i-2]:  
                count += 1  
        return count
```

3 - Notes

