

Problem 1930: Unique Length-3 Palindromic Subsequences

Problem Information

Difficulty: Medium

Acceptance Rate: 70.97%

Paid Only: No

Tags: Hash Table, String, Bit Manipulation, Prefix Sum

Problem Description

Given a string `s`, return _the number of**unique palindromes of length three** that are a **subsequence** of _`s`_.

Note that even if there are multiple ways to obtain the same subsequence, it is still only counted **once**.

A **palindrome** is a string that reads the same forwards and backwards.

A **subsequence** of a string is a new string generated from the original string with some characters (can be none) deleted without changing the relative order of the remaining characters.

* For example, `"ace"` is a subsequence of `"_a_ b _c_ d _e_ "`.

Example 1:

Input: s = "aabca" **Output:** 3 **Explanation:** The 3 palindromic subsequences of length 3 are: - "aba" (subsequence of "_a_ a _b_ c _a_ ") - "aaa" (subsequence of "_aa_ bc _a_ ") - "aca" (subsequence of "_a_ ab _ca_ ")

Example 2:

Input: s = "adc" **Output:** 0 **Explanation:** There are no palindromic subsequences of length 3 in "adc".

****Example 3:****

****Input:**** s = "bbcbaba" ****Output:**** 4 ****Explanation:**** The 4 palindromic subsequences of length 3 are: - "bbb" (subsequence of "_bb_ c _b_ aba") - "bcb" (subsequence of "_b_ b _cb_ aba") - "bab" (subsequence of "_b_ bcb _ab_ a") - "aba" (subsequence of "bbcb _aba_ ")

****Constraints:****

* `3 <= s.length <= 105` * `s` consists of only lowercase English letters.

Code Snippets

C++:

```
class Solution {
public:
    int countPalindromicSubsequence(string s) {
        }
    };
}
```

Java:

```
class Solution {
    public int countPalindromicSubsequence(String s) {
        }
    };
}
```

Python3:

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