

# 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:
```