

## 5809. Unique Length-3 Palindromic Subsequences

[My Submissions \(/contest/weekly-contest-249/problems/unique-length-3-palindromic-subsequences/submissions/\)](/contest/weekly-contest-249/problems/unique-length-3-palindromic-subsequences/submissions/)[Back to Contest \(/contest/weekly-contest-249/\)](/contest/weekly-contest-249/)

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 "abcde".

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Medium

### Example 1:

**Input:**  $s = \text{"aabca"}$

**Output:** 3

**Explanation:** The 3 palindromic subsequences of length 3 are:

- "aba" (subsequence of "aabca")
- "aaa" (subsequence of "aabca")
- "aca" (subsequence of "aabca")

### Example 2:

**Input:**  $s = \text{"adc"}$

**Output:** 0

**Explanation:** There are no palindromic subsequences of length 3 in "adc".

### Example 3:

**Input:**  $s = \text{"bbcbaba"}$

**Output:** 4

**Explanation:** The 4 palindromic subsequences of length 3 are:

- "bbb" (subsequence of "bbcbaba")
- "bcb" (subsequence of "bbcbaba")
- "bab" (subsequence of "bbcbaba")
- "aba" (subsequence of "bbcbaba")

### Constraints:

- $3 \leq s.length \leq 10^5$
- $s$  consists of only lowercase English letters.

JavaScript



```
1 const counter = (a_or_s) => { let map = new Map(); for (const i of a_or_s) map.set(i, map.get(i) + 1 || 1); return map; };
2 const countPalindromicSubsequence = (s) => {
3   let back = counter(s);
4   let front = new Map();
5   let se = new Set();
6   let n = s.length;
```

```
7  for (let i = 0; i < n; i++) {  
8      let mid = s[i];  
9      back.set(mid, (back.get(mid) || 0) - 1);  
10 for (let j = 0; j < 26; j++) {  
11     let c = String.fromCharCode(j + 97);  
12     if (front.get(c) && back.get(c)) se.add(c + mid + c);  
13 }  
14     front.set(mid, (front.get(mid) || 0) - 1);  
15 }  
16 return se.size;  
17 };
```

☐ Custom Testcase[Use Example Testcases](#)[Run](#)[Submit](#)**Submission Result: Accepted** (</submissions/detail/520619215/>) ?[More Details > \(/submissions/detail/520619215/\)](/submissions/detail/520619215/)

Share your acceptance!

&lt; 1

Copyright © 2021 LeetCode

[Help Center \(/support/\)](/support/) | [Jobs \(/jobs/\)](/jobs/) | [Bug Bounty \(/bugbounty/\)](/bugbounty/) | [Online Interview \(/interview/\)](/interview/) | [Students \(/student/\)](/student/) | [Terms \(/terms/\)](/terms/) |[Privacy Policy \(/privacy/\)](/privacy/) [United States \(/region/\)](/region/)