

Problem 2953: Count Complete Substrings

Problem Information

Difficulty: Hard

Acceptance Rate: 29.48%

Paid Only: No

Tags: Hash Table, String, Sliding Window

Problem Description

You are given a string `word` and an integer `k`.

A substring `s` of `word` is **complete** if:

- * Each character in `s` occurs **exactly** `k` times.
- * The difference between two adjacent characters is **at most** `2`. That is, for any two adjacent characters `c1` and `c2` in `s`, the absolute difference in their positions in the alphabet is **at most** `2`.

Return `_` the number of **complete** substrings of `word`.

A **substring** is a **non-empty** contiguous sequence of characters in a string.

Example 1:

Input: `word = "igigee", k = 2` **Output:** `3` **Explanation:** The complete substrings where each character appears exactly twice and the difference between adjacent characters is at most 2 are: `_igig_ee`, `igig_ee_`, `_igigee_`.

Example 2:

Input: `word = "aaabbbccc", k = 3` **Output:** `6` **Explanation:** The complete substrings where each character appears exactly three times and the difference between adjacent characters is at most 2 are: `_aaa_bbbccc`, `aaa_bbb_ccc`, `aaabbb_ccc_`, `_aaabbb_ccc`, `aaa_bbbccc_`, `_aaabbbccc_`.

Constraints:

* `1 <= word.length <= 105` * `word` consists only of lowercase English letters. * `1 <= k <= word.length`

Code Snippets

C++:

```
class Solution {  
public:  
    int countCompleteSubstrings(string word, int k) {  
  
    }  
};
```

Java:

```
class Solution {  
    public int countCompleteSubstrings(String word, int k) {  
  
    }  
}
```

Python3:

```
class Solution:  
    def countCompleteSubstrings(self, word: str, k: int) -> int:
```