

100145. Count Complete Substrings

[My Submissions \(/contest/weekly-contest-374/problems/count-complete-substrings/submissions/\)](#)

[Back to Contest \(/contest/weekly-contest-374/\)](#)

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.

User Accepted:	0
User Tried:	1
Total Accepted:	0
Total Submissions:	1
Difficulty:	Medium

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 "ig", "gi", and "ee".

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", "bbb", "ccc", "aabb", "bbcc", and "abcc".

Constraints:

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

JavaScript

📄 ↺ ⚙️

```
1 const addOneOrManyMap = (m, x, cnt = 1) => m.set(x, m.get(x) + cnt || cnt);
2 const removeOneOrManyMap = (m, x, cnt = 1) => { let occ = m.get(x); occ > cnt ? m.set(x, occ - cnt) : m.delete(x); };
3 const counter = (a_or_s) => { let m = new Map(); for (const x of a_or_s) m.set(x, m.get(x) + 1 || 1); return m; };
4 const ord = (c) => c.charCodeAt();
5
6 const countCompleteSubstrings = (s, k) => {
7   let l = 0, res = 0, n = s.length;
8   for (let i = 1; i < n; i++) {
9     if (Math.abs(ord(s[i]) - ord(s[i - 1])) > 2) {
10       res += cal(s.slice(l, i), k);
11       l = i;
12     }
13   }
14   res += cal(s.slice(l), k);
15   return res;
16 };
17
18
19 const cal = (s, k) => {
20   let res = 0, n = s.length;
```

# Start Now

## 2 Easy Steps

1. Click "[Start Now](#)"
2. Add Data Shield

```
27         removeOneOrManyMap(f, s[j]);
28         addOneOrManyMap(ff, f.get(s[j]));
29
30         removeOneOrManyMap(ff, f.get(s[j + 1]));
31         addOneOrManyMap(f, s[j + 1]);
32         addOneOrManyMap(ff, f.get(s[j + 1]));
33
34         if (ff.get(k) == i) res++;
35     }
36 }
37 return res;
38 };
```

☐ Custom Testcase[Use Example Testcases](#)[Run](#)[Submit](#)

Submission Result: **Accepted** (/submissions/detail/1115536369/) ?

[More Details > \(/submissions/detail/1115536369/\)](#)

Share your acceptance!

Copyright © 2023 LeetCode

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

 [United States \(/region\)](#)