Problem 3445: Maximum Difference Between Even and Odd Frequency II

Problem Information

Difficulty: Hard
Acceptance Rate: 48.72%
Paid Only: No
Tags: String, Sliding Window, Enumeration, Prefix Sum

Problem Description

You are given a string `s` and an integer `k`. Your task is to find the **maximum** difference between the frequency of **two** characters, `freq[a] - freq[b]`, in a substring `subs` of `s`, such that:

* `subs` has a size of **at least** `k`. * Character `a` has an _odd frequency_ in `subs`. * Character `b` has a **non-zero** _even frequency_ in `subs`.

Return the **maximum** difference.

Note that `subs` can contain more than 2 **distinct** characters.

Example 1:

Input: s = "12233", k = 4

Output: -1

Explanation:

For the substring `"12233"`, the frequency of `'1'` is 1 and the frequency of `'3'` is 2. The difference is `1 - 2 = -1`.

Example 2:

Input: s = "1122211", k = 3

Output: 1

Explanation:

For the substring `"11222"`, the frequency of `'2'` is 3 and the frequency of `'1'` is 2. The difference is `3 - 2 = 1`.

Example 3:

Input: s = "110", k = 3

Output: -1

Constraints:

* `3 <= s.length <= 3 * 104` * `s` consists only of digits `'0'` to `'4'`. * The input is generated that at least one substring has a character with an even frequency and a character with an odd frequency. * `1 <= k <= s.length`

Code Snippets

C++:

```
class Solution {
public:
int maxDifference(string s, int k) {

}
};
```

Java:

```
class Solution {
public int maxDifference(String s, int k) {

}
}
```

Python3:

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