

Problem 2982: Find Longest Special Substring That Occurs Thrice II

Problem Information

Difficulty: Medium

Acceptance Rate: 38.82%

Paid Only: No

Tags: Hash Table, String, Binary Search, Sliding Window, Counting

Problem Description

You are given a string `s` that consists of lowercase English letters.

A string is called **special** if it is made up of only a single character. For example, the string `"abc"` is not special, whereas the strings `"ddd"`, `"zz"`, and `"f"` are special.

Return the length of the **longest special substring** of `s` which occurs at least thrice, or `-1` if no special substring occurs at least thrice.

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

Example 1:

Input: `s = "aaaa"` **Output:** `2` **Explanation:** The longest special substring which occurs thrice is `"aa"`: substrings `"_**aa**_ aa"`, `"a _**aa**_ a"`, and `"aa _**aa**_ "`. It can be shown that the maximum length achievable is 2.

Example 2:

Input: `s = "abcdef"` **Output:** `-1` **Explanation:** There exists no special substring which occurs at least thrice. Hence return `-1`.

Example 3:

Input: s = "abcaba" **Output:** 1 **Explanation:** The longest special substring which occurs thrice is "a": substrings "_**a**_ bcaba", "abc _**a**_ ba", and "abcab _**a**_ ". It can be shown that the maximum length achievable is 1.

Constraints:

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

Code Snippets

C++:

```
class Solution {
public:
    int maximumLength(string s) {

    }
};
```

Java:

```
class Solution {
    public int maximumLength(String s) {

    }
}
```

Python3:

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