

# Problem 2981: Find Longest Special Substring That Occurs Thrice I

## Problem Information

**Difficulty:** Medium

**Acceptance Rate:** 61.91%

**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<sup>\*\*longest special substring\*\*</sup> of \_`s` \_which occurs<sup>\*\*at least thrice\*\*</sup>\_ , \_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 <= s.length <= 50` \* `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:
```