Problem 2981: Find Longest Special Substring That Occurs Thrice I

Problem Information

Difficulty: Medium
Acceptance Rate: 0.00%
Paid Only: No

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 <= 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:
```

Python:

```
class Solution(object):
def maximumLength(self, s):
"""
:type s: str
:rtype: int
"""
```

JavaScript:

```
/**
* @param {string} s
* @return {number}
*/
var maximumLength = function(s) {

};
```

TypeScript:

```
function maximumLength(s: string): number {

};
```

C#:

```
public class Solution {
public int MaximumLength(string s) {

}
}
```

C:

```
int maximumLength(char* s) {

}
```

Go:

```
func maximumLength(s string) int {

}
```

Kotlin:

```
class Solution {
fun maximumLength(s: String): Int {

}
}
```

Swift:

```
class Solution {
func maximumLength(_ s: String) -> Int {

}
}
```

Rust:

```
impl Solution {
pub fn maximum_length(s: String) -> i32 {

}
}
```

Ruby:

```
# @param {String} s
# @return {Integer}
def maximum_length(s)

end
```

PHP:

```
class Solution {

/**
* @param String $s
* @return Integer
*/
function maximumLength($s) {

}
}
```

Dart:

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

}
}
```

Scala:

```
object Solution {
def maximumLength(s: String): Int = {

}
}
```

Elixir:

```
defmodule Solution do
@spec maximum_length(s :: String.t) :: integer
def maximum_length(s) do
```

```
end
end
```

Erlang:

```
-spec maximum_length(S :: unicode:unicode_binary()) -> integer().
maximum_length(S) ->
.
```

Racket:

```
(define/contract (maximum-length s)
(-> string? exact-integer?)
)
```

Solutions

C++ Solution:

```
/*
* Problem: Find Longest Special Substring That Occurs Thrice I
* Difficulty: Medium
* Tags: array, string, tree, hash, search
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

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

}
};
```

Java Solution:

```
/**
* Problem: Find Longest Special Substring That Occurs Thrice I
* Difficulty: Medium
* Tags: array, string, tree, hash, search
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

class Solution {
public int maximumLength(String s) {

}
}
```

Python3 Solution:

```
"""
Problem: Find Longest Special Substring That Occurs Thrice I
Difficulty: Medium
Tags: array, string, tree, hash, search

Approach: Use two pointers or sliding window technique
Time Complexity: O(n) or O(n log n)
Space Complexity: O(h) for recursion stack where h is height
"""

class Solution:
def maximumLength(self, s: str) -> int:
# TODO: Implement optimized solution
pass
```

Python Solution:

```
class Solution(object):
def maximumLength(self, s):
"""
:type s: str
:rtype: int
"""
```

JavaScript Solution:

```
/**
* Problem: Find Longest Special Substring That Occurs Thrice I
* Difficulty: Medium
* Tags: array, string, tree, hash, search
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

/**
* @param {string} s
* @return {number}
*/
var maximumLength = function(s) {

};
```

TypeScript Solution:

```
/**
* Problem: Find Longest Special Substring That Occurs Thrice I
* Difficulty: Medium
* Tags: array, string, tree, hash, search
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

function maximumLength(s: string): number {

};
```

C# Solution:

```
/*
* Problem: Find Longest Special Substring That Occurs Thrice I
* Difficulty: Medium
* Tags: array, string, tree, hash, search
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

public class Solution {
public int MaximumLength(string s) {

}
}
```

C Solution:

```
/*
* Problem: Find Longest Special Substring That Occurs Thrice I
* Difficulty: Medium
* Tags: array, string, tree, hash, search
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

int maximumLength(char* s) {

}
```

Go Solution:

```
// Problem: Find Longest Special Substring That Occurs Thrice I
// Difficulty: Medium
// Tags: array, string, tree, hash, search
//
// Approach: Use two pointers or sliding window technique
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(h) for recursion stack where h is height

func maximumLength(s string) int {

}
```

Kotlin Solution:

```
class Solution {
fun maximumLength(s: String): Int {

}
}
```

Swift Solution:

```
class Solution {
func maximumLength(_ s: String) -> Int {

}
}
```

Rust Solution:

```
// Problem: Find Longest Special Substring That Occurs Thrice I
// Difficulty: Medium
// Tags: array, string, tree, hash, search
//
// Approach: Use two pointers or sliding window technique
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(h) for recursion stack where h is height

impl Solution {
pub fn maximum_length(s: String) -> i32 {

}
}
```

Ruby Solution:

```
# @param {String} s
# @return {Integer}
def maximum_length(s)

end
```

PHP Solution:

```
class Solution {
```

```
/**
* @param String $s
* @return Integer
*/
function maximumLength($s) {

}
}
```

Dart Solution:

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

}
}
```

Scala Solution:

```
object Solution {
def maximumLength(s: String): Int = {

}
}
```

Elixir Solution:

```
defmodule Solution do
@spec maximum_length(s :: String.t) :: integer
def maximum_length(s) do

end
end
```

Erlang Solution:

```
-spec maximum_length(S :: unicode:unicode_binary()) -> integer().
maximum_length(S) ->
.
```

Racket Solution:

```
(define/contract (maximum-length s)
(-> string? exact-integer?)
)
```