Problem 2730: Find the Longest Semi-Repetitive Substring

Problem Information

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

Problem Description

You are given a digit string

s

that consists of digits from 0 to 9.

A string is called

semi-repetitive

if there is

at most

one adjacent pair of the same digit. For example,

"0010"

,

"002020"

,

"0123"

,

"2002"

, and

"54944"

are semi-repetitive while the following are not:

"00101022"

(adjacent same digit pairs are 00 and 22), and

"1101234883"

(adjacent same digit pairs are 11 and 88).

Return the length of the

longest semi-repetitive

substring

of

s

.

Example 1:

Input:

s = "52233"

Output:

4

Explanation:

The longest semi-repetitive substring is "5223". Picking the whole string "52233" has two adjacent same digit pairs 22 and 33, but at most one is allowed.

Example 2:

Input:

s = "5494"

Output:

4

Explanation:

s

is a semi-repetitive string.

Example 3:

Input:

s = "1111111"

Output:

2

Explanation:

The longest semi-repetitive substring is "11". Picking the substring "111" has two adjacent same digit pairs, but at most one is allowed.

Constraints:

1 <= s.length <= 50

'0' <= s[i] <= '9'

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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

Python:

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

JavaScript:

```
/**
* @param {string} s
```

```
* @return {number}
*/
var longestSemiRepetitiveSubstring = function(s) {

};
```

TypeScript:

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

};
```

C#:

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

}
}
```

C:

```
int longestSemiRepetitiveSubstring(char* s) {

}
```

Go:

```
func longestSemiRepetitiveSubstring(s string) int {

}
```

Kotlin:

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

}
}
```

Swift:

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

}
}
```

Rust:

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

}
}
```

Ruby:

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

end
```

PHP:

```
class Solution {

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

}
}
```

Dart:

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

}
}
```

Scala:

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

}
}
```

Elixir:

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

end
end
```

Erlang:

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

Racket:

```
(define/contract (longest-semi-repetitive-substring s)
(-> string? exact-integer?)
)
```

Solutions

C++ Solution:

```
/*
* Problem: Find the Longest Semi-Repetitive Substring
* Difficulty: Medium
* Tags: array, string, tree
*
* 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 longestSemiRepetitiveSubstring(string s) {

}
};
```

Java Solution:

```
/**
* Problem: Find the Longest Semi-Repetitive Substring
* Difficulty: Medium
* Tags: array, string, tree
*
* 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 longestSemiRepetitiveSubstring(String s) {

}
}
```

Python3 Solution:

```
"""
Problem: Find the Longest Semi-Repetitive Substring
Difficulty: Medium
Tags: array, string, tree

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 longestSemiRepetitiveSubstring(self, s: str) -> int:
```

```
# TODO: Implement optimized solution
pass
```

Python Solution:

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

JavaScript Solution:

```
/**
* Problem: Find the Longest Semi-Repetitive Substring
* Difficulty: Medium
* Tags: array, string, tree
*
* 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 longestSemiRepetitiveSubstring = function(s) {

};
```

TypeScript Solution:

```
/**
* Problem: Find the Longest Semi-Repetitive Substring
* Difficulty: Medium
* Tags: array, string, tree
*
* 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 longestSemiRepetitiveSubstring(s: string): number {

};
```

C# Solution:

```
/*
* Problem: Find the Longest Semi-Repetitive Substring
* Difficulty: Medium
* Tags: array, string, tree
*
* 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 LongestSemiRepetitiveSubstring(string s) {

}
}
```

C Solution:

```
/*
* Problem: Find the Longest Semi-Repetitive Substring
* Difficulty: Medium
* Tags: array, string, tree
*
* 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 longestSemiRepetitiveSubstring(char* s) {

}
```

Go Solution:

```
// Problem: Find the Longest Semi-Repetitive Substring
// Difficulty: Medium
// Tags: array, string, tree
//
// 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 longestSemiRepetitiveSubstring(s string) int {

}
```

Kotlin Solution:

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

}
}
```

Swift Solution:

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

}
}
```

Rust Solution:

```
// Problem: Find the Longest Semi-Repetitive Substring
// Difficulty: Medium
// Tags: array, string, tree
//
// 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 longest_semi_repetitive_substring(s: String) -> i32 {
```

```
}
}
```

Ruby Solution:

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

end
```

PHP Solution:

```
class Solution {

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

}
}
```

Dart Solution:

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

}
}
```

Scala Solution:

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

}
}
```

Elixir Solution:

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

end
end
```

Erlang Solution:

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

Racket Solution:

```
(define/contract (longest-semi-repetitive-substring s)
(-> string? exact-integer?)
)
```