Problem 1062: Longest Repeating Substring

Problem Information

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

Problem Description

Given a string

s

, return

the length of the longest repeating substrings

. If no repeating substring exists, return

0

.

Example 1:

Input:

s = "abcd"

Output:

0

Explanation:

There is no repeating substring.

Example 2:

Input:

s = "abbaba"

Output:

2

Explanation:

The longest repeating substrings are "ab" and "ba", each of which occurs twice.

Example 3:

Input:

s = "aabcaabdaab"

Output:

3

Explanation:

The longest repeating substring is "aab", which occurs

3

times.

Constraints:

1 <= s.length <= 2000

s

consists of lowercase English letters.

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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

Python:

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

JavaScript:

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

```
var longestRepeatingSubstring = function(s) {

};
```

TypeScript:

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

};
```

C#:

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

}
}
```

C:

```
int longestRepeatingSubstring(char* s) {

}
```

Go:

```
func longestRepeatingSubstring(s string) int {

}
```

Kotlin:

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

}
}
```

Swift:

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

```
}
}
```

Rust:

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

}
}
```

Ruby:

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

end
```

PHP:

```
class Solution {

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

}
}
```

Dart:

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

}
}
```

Scala:

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

}
}
```

Elixir:

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

end
end
```

Erlang:

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

Racket:

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

Solutions

C++ Solution:

```
/*
* Problem: Longest Repeating Substring
* Difficulty: Medium
* Tags: array, string, tree, dp, hash, search
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(n) or O(n * m) for DP table
*/
```

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

}
};
```

Java Solution:

```
/**
* Problem: Longest Repeating Substring
* Difficulty: Medium
* Tags: array, string, tree, dp, hash, search
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(n) or O(n * m) for DP table
*/

class Solution {
public int longestRepeatingSubstring(String s) {

}
}
```

Python3 Solution:

```
"""
Problem: Longest Repeating Substring
Difficulty: Medium
Tags: array, string, tree, dp, hash, search

Approach: Use two pointers or sliding window technique
Time Complexity: O(n) or O(n log n)
Space Complexity: O(n) or O(n * m) for DP table
"""

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

```
pass
```

Python Solution:

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

JavaScript Solution:

```
/**
* Problem: Longest Repeating Substring
* Difficulty: Medium
* Tags: array, string, tree, dp, hash, search
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(n) or O(n * m) for DP table
*/

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

};
```

TypeScript Solution:

```
/**
* Problem: Longest Repeating Substring
* Difficulty: Medium
* Tags: array, string, tree, dp, hash, search
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(n) or O(n * m) for DP table
```

```
*/

function longestRepeatingSubstring(s: string): number {

};
```

C# Solution:

```
/*
* Problem: Longest Repeating Substring
* Difficulty: Medium
* Tags: array, string, tree, dp, hash, search
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(n) or O(n * m) for DP table
*/

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

}
}
```

C Solution:

```
/*
* Problem: Longest Repeating Substring
* Difficulty: Medium
* Tags: array, string, tree, dp, hash, search
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(n) or O(n * m) for DP table
*/

int longestRepeatingSubstring(char* s) {

}
```

Go Solution:

```
// Problem: Longest Repeating Substring
// Difficulty: Medium
// Tags: array, string, tree, dp, hash, search
//
// Approach: Use two pointers or sliding window technique
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(n) or O(n * m) for DP table

func longestRepeatingSubstring(s string) int {

}
```

Kotlin Solution:

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

}
}
```

Swift Solution:

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

}
}
```

Rust Solution:

```
// Problem: Longest Repeating Substring
// Difficulty: Medium
// Tags: array, string, tree, dp, hash, search
//
// Approach: Use two pointers or sliding window technique
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(n) or O(n * m) for DP table

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

}
```

```
}
```

Ruby Solution:

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

end
```

PHP Solution:

```
class Solution {

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

}
}
```

Dart Solution:

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

}
}
```

Scala Solution:

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

}
}
```

Elixir Solution:

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

end
end
```

Erlang Solution:

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

Racket Solution:

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