Problem 1461: Check If a String Contains All Binary Codes of Size K

Problem Information

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

Problem Description

Given a binary string

s

and an integer

k

, return

true

if every binary code of length

k

is a substring of

s

. Otherwise, return

false

.

Example 1:

Input:

s = "00110110", k = 2

Output:

true

Explanation:

The binary codes of length 2 are "00", "01", "10" and "11". They can be all found as substrings at indices 0, 1, 3 and 2 respectively.

Example 2:

Input:

s = "0110", k = 1

Output:

true

Explanation:

The binary codes of length 1 are "0" and "1", it is clear that both exist as a substring.

Example 3:

Input:

s = "0110", k = 2

Output:

false

Explanation:

The binary code "00" is of length 2 and does not exist in the array.

Constraints:

1 <= s.length <= 5 * 10

5

s[i]

is either

'0'

or

'1'

.

1 <= k <= 20

Code Snippets

C++:

```
class Solution {
public:
bool hasAllCodes(string s, int k) {

}
};
```

Java:

```
class Solution {
public boolean hasAllCodes(String s, int k) {



}
}
```

Python3:

```
class Solution:
def hasAllCodes(self, s: str, k: int) -> bool:
```

Python:

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

JavaScript:

```
/**
* @param {string} s
* @param {number} k
* @return {boolean}
*/
var hasAllCodes = function(s, k) {

};
```

TypeScript:

```
function hasAllCodes(s: string, k: number): boolean {

};
```

C#:

```
public class Solution {
public bool HasAllCodes(string s, int k) {

}

}
```

C:

```
bool hasAllCodes(char* s, int k) {

}
```

Go:

```
func hasAllCodes(s string, k int) bool {

}
```

Kotlin:

```
class Solution {
fun hasAllCodes(s: String, k: Int): Boolean {

}
}
```

Swift:

```
class Solution {
func hasAllCodes(_ s: String, _ k: Int) -> Bool {

}
}
```

Rust:

```
impl Solution {
pub fn has_all_codes(s: String, k: i32) -> bool {

}
}
```

Ruby:

```
# @param {String} s
# @param {Integer} k
# @return {Boolean}
def has_all_codes(s, k)

end
```

PHP:

```
class Solution {

/**
* @param String $s
* @param Integer $k
* @return Boolean
*/
function hasAllCodes($s, $k) {

}
}
```

Dart:

```
class Solution {
bool hasAllCodes(String s, int k) {

}
}
```

Scala:

```
object Solution {
def hasAllCodes(s: String, k: Int): Boolean = {

}
}
```

Elixir:

```
defmodule Solution do
@spec has_all_codes(s :: String.t, k :: integer) :: boolean
def has_all_codes(s, k) do

end
```

```
end
```

Erlang:

```
-spec has_all_codes(S :: unicode:unicode_binary(), K :: integer()) ->
boolean().
has_all_codes(S, K) ->
.
```

Racket:

```
(define/contract (has-all-codes s k)
(-> string? exact-integer? boolean?)
)
```

Solutions

C++ Solution:

```
/*
* Problem: Check If a String Contains All Binary Codes of Size K
* Difficulty: Medium
* Tags: array, string, tree, hash
*
* 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:
bool hasAllCodes(string s, int k) {

}
};
```

Java Solution:

```
/**
* Problem: Check If a String Contains All Binary Codes of Size K
* Difficulty: Medium
* Tags: array, string, tree, hash
*
* 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 boolean hasAllCodes(String s, int k) {

}
}
```

Python3 Solution:

```
"""
Problem: Check If a String Contains All Binary Codes of Size K
Difficulty: Medium
Tags: array, string, tree, hash

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 hasAllCodes(self, s: str, k: int) -> bool:
# TODO: Implement optimized solution
pass
```

Python Solution:

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

JavaScript Solution:

```
/**
* Problem: Check If a String Contains All Binary Codes of Size K
* Difficulty: Medium
* Tags: array, string, tree, hash
*
* 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
* @param {number} k
* @return {boolean}
*/
var hasAllCodes = function(s, k) {

};
```

TypeScript Solution:

```
/**
* Problem: Check If a String Contains All Binary Codes of Size K
* Difficulty: Medium
* Tags: array, string, tree, hash
*
* 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 hasAllCodes(s: string, k: number): boolean {

};
```

C# Solution:

```
/*
* Problem: Check If a String Contains All Binary Codes of Size K
* Difficulty: Medium
* Tags: array, string, tree, hash
```

```
*
* 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 bool HasAllCodes(string s, int k) {

}
}
```

C Solution:

```
/*
* Problem: Check If a String Contains All Binary Codes of Size K
* Difficulty: Medium
* Tags: array, string, tree, hash
*
* 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
*/

bool hasAllCodes(char* s, int k) {

}
```

Go Solution:

```
// Problem: Check If a String Contains All Binary Codes of Size K
// Difficulty: Medium
// Tags: array, string, tree, hash
//
// 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 hasAllCodes(s string, k int) bool {

}
```

Kotlin Solution:

```
class Solution {
fun hasAllCodes(s: String, k: Int): Boolean {

}
}
```

Swift Solution:

```
class Solution {
func hasAllCodes(_ s: String, _ k: Int) -> Bool {

}
}
```

Rust Solution:

```
// Problem: Check If a String Contains All Binary Codes of Size K
// Difficulty: Medium
// Tags: array, string, tree, hash
//
// 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 has_all_codes(s: String, k: i32) -> bool {

}
}
```

Ruby Solution:

```
# @param {String} s
# @param {Integer} k
# @return {Boolean}
def has_all_codes(s, k)

end
```

PHP Solution:

```
class Solution {

/**
* @param String $s
* @param Integer $k
* @return Boolean
*/
function hasAllCodes($s, $k) {

}
}
```

Dart Solution:

```
class Solution {
bool hasAllCodes(String s, int k) {

}
}
```

Scala Solution:

```
object Solution {
def hasAllCodes(s: String, k: Int): Boolean = {

}
}
```

Elixir Solution:

```
defmodule Solution do
@spec has_all_codes(s :: String.t, k :: integer) :: boolean
def has_all_codes(s, k) do

end
end
```

Erlang Solution:

```
-spec has_all_codes(S :: unicode:unicode_binary(), K :: integer()) ->
boolean().
has_all_codes(S, K) ->
.
```

Racket Solution:

```
(define/contract (has-all-codes s k)
(-> string? exact-integer? boolean?)
)
```