Problem 3084: Count Substrings Starting and Ending with Given Character

Problem Information

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

Problem Description

You are given a string

s

and a character

c

. Return

the total number of

substrings

of

s

that start and end with

c

.

Example 1:

Input:

s = "abada", c = "a"

Output:

6

Explanation:

Substrings starting and ending with

"a"

are:

"

a

bada"

,

"

aba

da"

,

"

abada

"

,

"ab

a

da"

,

"ab

ada

"

,

"abad

a

"

.

Example 2:

Input:

s = "zzz", c = "z"

Output:

6

Explanation:

There are a total of

6

substrings in

s

and all start and end with

"z"

.

Constraints:

1 <= s.length <= 10

5

s

and

c

consist only of lowercase English letters.

Code Snippets

C++:

```
class Solution {
public:
long long countSubstrings(string s, char c) {

}
};
```

Java:

```
class Solution {
public long countSubstrings(String s, char c) {

}
}
```

Python3:

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

Python:

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

JavaScript:

```
/**
* @param {string} s
* @param {character} c
* @return {number}
*/
var countSubstrings = function(s, c) {

};
```

TypeScript:

```
function countSubstrings(s: string, c: string): number {

};
```

C#:

```
public class Solution {
public long CountSubstrings(string s, char c) {
}
}
```

C:

```
long long countSubstrings(char* s, char c) {

}
```

Go:

```
func countSubstrings(s string, c byte) int64 {

}
```

Kotlin:

```
class Solution {
fun countSubstrings(s: String, c: Char): Long {

}
}
```

Swift:

```
class Solution {
func countSubstrings(_ s: String, _ c: Character) -> Int {

}
}
```

Rust:

```
impl Solution {
pub fn count_substrings(s: String, c: char) -> i64 {

}
}
```

Ruby:

```
# @param {String} s
# @param {Character} c
# @return {Integer}
def count_substrings(s, c)

end
```

PHP:

```
class Solution {

/**
* @param String $s
* @param String $c
* @return Integer
*/
function countSubstrings($s, $c) {

}
}
```

Dart:

```
class Solution {
int countSubstrings(String s, String c) {

}
}
```

Scala:

```
object Solution {
def countSubstrings(s: String, c: Char): Long = {

}
}
```

Elixir:

```
defmodule Solution do
@spec count_substrings(s :: String.t, c :: char) :: integer
def count_substrings(s, c) do
```

```
end
end
```

Erlang:

```
-spec count_substrings(S :: unicode:unicode_binary(), C :: char()) ->
integer().
count_substrings(S, C) ->
.
```

Racket:

```
(define/contract (count-substrings s c)
(-> string? char? exact-integer?)
)
```

Solutions

C++ Solution:

```
/*
* Problem: Count Substrings Starting and Ending with Given Character
* Difficulty: Medium
* Tags: string, tree, math
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

class Solution {
public:
long long countSubstrings(string s, char c) {

}
};
```

Java Solution:

```
/**
* Problem: Count Substrings Starting and Ending with Given Character
* Difficulty: Medium
* Tags: string, tree, math
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

class Solution {
public long countSubstrings(String s, char c) {

}
}
```

Python3 Solution:

```
"""
Problem: Count Substrings Starting and Ending with Given Character
Difficulty: Medium
Tags: string, tree, math

Approach: String manipulation with hash map or two pointers
Time Complexity: O(n) or O(n log n)
Space Complexity: O(h) for recursion stack where h is height
"""

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

Python Solution:

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

JavaScript Solution:

```
/**
* Problem: Count Substrings Starting and Ending with Given Character
* Difficulty: Medium
* Tags: string, tree, math
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

/**
* @param {string} s
* @param {character} c
* @return {number}
*/
var countSubstrings = function(s, c) {

};
```

TypeScript Solution:

```
/**
* Problem: Count Substrings Starting and Ending with Given Character
* Difficulty: Medium
* Tags: string, tree, math
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

function countSubstrings(s: string, c: string): number {

};
```

C# Solution:

```
/*
* Problem: Count Substrings Starting and Ending with Given Character
* Difficulty: Medium
* Tags: string, tree, math
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

public class Solution {
public long CountSubstrings(string s, char c) {

}
}
```

C Solution:

```
/*
* Problem: Count Substrings Starting and Ending with Given Character
* Difficulty: Medium
* Tags: string, tree, math
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

long long countSubstrings(char* s, char c) {

}
```

Go Solution:

```
// Problem: Count Substrings Starting and Ending with Given Character
// Difficulty: Medium
// Tags: string, tree, math
//
// Approach: String manipulation with hash map or two pointers
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(h) for recursion stack where h is height

func countSubstrings(s string, c byte) int64 {
}
```

Kotlin Solution:

```
class Solution {
fun countSubstrings(s: String, c: Char): Long {

}
}
```

Swift Solution:

```
class Solution {
func countSubstrings(_ s: String, _ c: Character) -> Int {

}
}
```

Rust Solution:

```
// Problem: Count Substrings Starting and Ending with Given Character
// Difficulty: Medium
// Tags: string, tree, math
//
// Approach: String manipulation with hash map or two pointers
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(h) for recursion stack where h is height

impl Solution {
pub fn count_substrings(s: String, c: char) -> i64 {

}
}
```

Ruby Solution:

```
# @param {String} s
# @param {Character} c
# @return {Integer}
def count_substrings(s, c)
```

```
end
```

PHP Solution:

```
class Solution {

/**
* @param String $s
* @param String $c
* @return Integer
*/
function countSubstrings($s, $c) {

}
}
```

Dart Solution:

```
class Solution {
int countSubstrings(String s, String c) {

}
}
```

Scala Solution:

```
object Solution {
def countSubstrings(s: String, c: Char): Long = {

}
}
```

Elixir Solution:

```
defmodule Solution do
@spec count_substrings(s :: String.t, c :: char) :: integer
def count_substrings(s, c) do

end
end
```

Erlang Solution:

```
-spec count_substrings(S :: unicode:unicode_binary(), C :: char()) ->
integer().
count_substrings(S, C) ->
.
```

Racket Solution:

```
(define/contract (count-substrings s c)
(-> string? char? exact-integer?)
)
```