Problem 28: Find the Index of the First Occurrence in a String

Problem Information

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

Problem Description

Given two strings

needle

and

haystack

, return the index of the first occurrence of

needle

in

haystack

, or

-1

if

needle

is not part of

haystack

.

Example 1:

Input:

haystack = "sadbutsad", needle = "sad"

Output:

0

Explanation:

"sad" occurs at index 0 and 6. The first occurrence is at index 0, so we return 0.

Example 2:

Input:

haystack = "leetcode", needle = "leeto"

Output:

-1

Explanation:

"leeto" did not occur in "leetcode", so we return -1.

Constraints:

1 <= haystack.length, needle.length <= 10

4

haystack

and

needle

consist of only lowercase English characters.

Code Snippets

C++:

```
class Solution {
public:
int strStr(string haystack, string needle) {

}
};
```

Java:

```
class Solution {
public int strStr(String haystack, String needle) {

}
}
```

Python3:

```
class Solution:
def strStr(self, haystack: str, needle: str) -> int:
```

Python:

```
class Solution(object):
def strStr(self, haystack, needle):
"""
:type haystack: str
:type needle: str
:rtype: int
"""
```

JavaScript:

```
/**
* @param {string} haystack
* @param {string} needle
* @return {number}
*/
var strStr = function(haystack, needle) {

};
```

TypeScript:

```
function strStr(haystack: string, needle: string): number {

};
```

C#:

```
public class Solution {
public int StrStr(string haystack, string needle) {

}
}
```

C:

```
int strStr(char* haystack, char* needle) {

}
```

Go:

```
func strStr(haystack string, needle string) int {

}
```

Kotlin:

```
class Solution {
fun strStr(haystack: String, needle: String): Int {

}
}
```

Swift:

```
class Solution {
func strStr(_ haystack: String, _ needle: String) -> Int {

}
}
```

Rust:

```
impl Solution {
pub fn str_str(haystack: String, needle: String) -> i32 {

}
}
```

Ruby:

```
# @param {String} haystack
# @param {String} needle
# @return {Integer}
def str_str(haystack, needle)

end
```

PHP:

```
class Solution {

/**
* @param String $haystack
* @param String $needle
* @return Integer
*/
function strStr($haystack, $needle) {

}
}
```

Dart:

```
class Solution {
int strStr(String haystack, String needle) {

}
}
```

Scala:

```
object Solution {
def strStr(haystack: String, needle: String): Int = {

}
}
```

Elixir:

```
defmodule Solution do
@spec str_str(haystack :: String.t, needle :: String.t) :: integer
def str_str(haystack, needle) do

end
end
```

Erlang:

```
-spec str_str(Haystack :: unicode:unicode_binary(), Needle ::
unicode:unicode_binary()) -> integer().
str_str(Haystack, Needle) ->
.
```

Racket:

```
(define/contract (str-str haystack needle)
(-> string? string? exact-integer?)
)
```

Solutions

C++ Solution:

```
/*
* Problem: Find the Index of the First Occurrence in a String
* Difficulty: Easy
* Tags: array, string, stack
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

class Solution {
public:
int strStr(string haystack, string needle) {

}
};
```

Java Solution:

```
/**
* Problem: Find the Index of the First Occurrence in a String
* Difficulty: Easy
* Tags: array, string, stack
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

class Solution {
public int strStr(String haystack, String needle) {

}
}
```

Python3 Solution:

```
"""
Problem: Find the Index of the First Occurrence in a String
Difficulty: Easy
Tags: array, string, stack
```

```
Approach: Use two pointers or sliding window technique
Time Complexity: O(n) or O(n log n)
Space Complexity: O(1) to O(n) depending on approach
"""

class Solution:
def strStr(self, haystack: str, needle: str) -> int:
# TODO: Implement optimized solution
pass
```

Python Solution:

```
class Solution(object):
def strStr(self, haystack, needle):
"""
:type haystack: str
:type needle: str
:rtype: int
"""
```

JavaScript Solution:

```
/**
* Problem: Find the Index of the First Occurrence in a String
* Difficulty: Easy
* Tags: array, string, stack
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

/**
* @param {string} haystack
* @param {string} needle
* @return {number}
*/
var strStr = function(haystack, needle) {

};
```

TypeScript Solution:

```
/**
* Problem: Find the Index of the First Occurrence in a String
* Difficulty: Easy
* Tags: array, string, stack
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

function strStr(haystack: string, needle: string): number {

};
```

C# Solution:

```
/*
* Problem: Find the Index of the First Occurrence in a String
* Difficulty: Easy
* Tags: array, string, stack
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

public class Solution {
public int StrStr(string haystack, string needle) {

}
}
```

C Solution:

```
/*
* Problem: Find the Index of the First Occurrence in a String
* Difficulty: Easy
* Tags: array, string, stack
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

int strStr(char* haystack, char* needle) {

}
```

Go Solution:

```
// Problem: Find the Index of the First Occurrence in a String
// Difficulty: Easy
// Tags: array, string, stack
//
// Approach: Use two pointers or sliding window technique
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(1) to O(n) depending on approach

func strStr(haystack string, needle string) int {

}
```

Kotlin Solution:

```
class Solution {
fun strStr(haystack: String, needle: String): Int {

}
}
```

Swift Solution:

```
class Solution {
func strStr(_ haystack: String, _ needle: String) -> Int {

}
}
```

Rust Solution:

```
// Problem: Find the Index of the First Occurrence in a String
// Difficulty: Easy
// Tags: array, string, stack
//
// Approach: Use two pointers or sliding window technique
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(1) to O(n) depending on approach

impl Solution {
pub fn str_str(haystack: String, needle: String) -> i32 {

}
}
```

Ruby Solution:

```
# @param {String} haystack
# @param {String} needle
# @return {Integer}
def str_str(haystack, needle)

end
```

PHP Solution:

```
class Solution {

/**
* @param String $haystack
* @param String $needle
* @return Integer
*/
function strStr($haystack, $needle) {

}
}
```

Dart Solution:

```
class Solution {
int strStr(String haystack, String needle) {

}
}
```

Scala Solution:

```
object Solution {
def strStr(haystack: String, needle: String): Int = {

}
}
```

Elixir Solution:

```
defmodule Solution do
@spec str_str(haystack :: String.t, needle :: String.t) :: integer
def str_str(haystack, needle) do

end
end
```

Erlang Solution:

```
-spec str_str(Haystack :: unicode:unicode_binary(), Needle ::
unicode:unicode_binary()) -> integer().
str_str(Haystack, Needle) ->
.
```

Racket Solution:

```
(define/contract (str-str haystack needle)
(-> string? string? exact-integer?)
)
```