Problem 3628: Maximum Number of Subsequences After One Inserting

Problem Information

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

Problem Description

You are given a string

s

consisting of uppercase English letters.

You are allowed to insert

at most one

uppercase English letter at

any

position (including the beginning or end) of the string.

Return the

maximum

number of

"LCT"

subsequences

that can be formed in the resulting string after

at most one insertion

.

Example 1:

Input:

s = "LMCT"

Output:

2

Explanation:

We can insert a

"L"

at the beginning of the string s to make

"LLMCT"

, which has 2 subsequences, at indices [0, 3, 4] and [1, 3, 4].

Example 2:

Input:

s = "LCCT"

Output:

4

Explanation:

We can insert a

"L"

at the beginning of the string s to make

"LLCCT"

, which has 4 subsequences, at indices [0, 2, 4], [0, 3, 4], [1, 2, 4] and [1, 3, 4].

Example 3:

Input:

s = "L"

Output:

0

Explanation:

Since it is not possible to obtain the subsequence

"LCT"

by inserting a single letter, the result is 0.

Constraints:

1 <= s.length <= 10

5

s

consists of uppercase English letters.

Code Snippets

C++:

```
class Solution {
public:
long long numOfSubsequences(string s) {

}
};
```

Java:

```
class Solution {
public long numOfSubsequences(String s) {

}
}
```

Python3:

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

Python:

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

JavaScript:

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

```
};
```

TypeScript:

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

};
```

C#:

```
public class Solution {
public long NumOfSubsequences(string s) {

}
}
```

C:

```
long long numOfSubsequences(char* s) {

}
```

Go:

```
func numOfSubsequences(s string) int64 {

}
```

Kotlin:

```
class Solution {
fun numOfSubsequences(s: String): Long {

}
}
```

Swift:

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

}
```

```
}
```

Rust:

```
impl Solution {
pub fn num_of_subsequences(s: String) -> i64 {

}
}
```

Ruby:

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

end
```

PHP:

```
class Solution {

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

}
}
```

Dart:

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

}
}
```

Scala:

```
object Solution {
def numOfSubsequences(s: String): Long = {

}
}
```

Elixir:

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

end
end
```

Erlang:

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

Racket:

```
(define/contract (num-of-subsequences s)
(-> string? exact-integer?)
)
```

Solutions

C++ Solution:

```
/*
* Problem: Maximum Number of Subsequences After One Inserting
* Difficulty: Medium
* Tags: array, string, dp, greedy
*
* 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:
long long numOfSubsequences(string s) {

}
};
```

Java Solution:

```
/**
* Problem: Maximum Number of Subsequences After One Inserting
* Difficulty: Medium
* Tags: array, string, dp, greedy
*
* 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 long numOfSubsequences(String s) {

}
}
```

Python3 Solution:

```
"""
Problem: Maximum Number of Subsequences After One Inserting
Difficulty: Medium
Tags: array, string, dp, greedy

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

Python Solution:

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

JavaScript Solution:

```
/**
* Problem: Maximum Number of Subsequences After One Inserting
* Difficulty: Medium
* Tags: array, string, dp, greedy
*
* 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 numOfSubsequences = function(s) {

};
```

TypeScript Solution:

```
/**
* Problem: Maximum Number of Subsequences After One Inserting
* Difficulty: Medium
* Tags: array, string, dp, greedy
*
* 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 numOfSubsequences(s: string): number {
```

```
};
```

C# Solution:

```
/*
* Problem: Maximum Number of Subsequences After One Inserting
* Difficulty: Medium
* Tags: array, string, dp, greedy
*
* 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 long NumOfSubsequences(string s) {

}
}
```

C Solution:

```
/*
* Problem: Maximum Number of Subsequences After One Inserting
* Difficulty: Medium
* Tags: array, string, dp, greedy
*
* 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
*/

long long numOfSubsequences(char* s) {

}
```

Go Solution:

```
// Problem: Maximum Number of Subsequences After One Inserting
// Difficulty: Medium
```

```
// Tags: array, string, dp, greedy
//
// 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 numOfSubsequences(s string) int64 {

}
```

Kotlin Solution:

```
class Solution {
fun numOfSubsequences(s: String): Long {

}
}
```

Swift Solution:

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

}
}
```

Rust Solution:

```
// Problem: Maximum Number of Subsequences After One Inserting
// Difficulty: Medium
// Tags: array, string, dp, greedy
//
// 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 num_of_subsequences(s: String) -> i64 {

}
}
```

Ruby Solution:

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

end
```

PHP Solution:

```
class Solution {

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

}
}
```

Dart Solution:

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

}
}
```

Scala Solution:

```
object Solution {
def numOfSubsequences(s: String): Long = {

}
}
```

Elixir Solution:

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

```
end
end
```

Erlang Solution:

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

Racket Solution:

```
(define/contract (num-of-subsequences s)
(-> string? exact-integer?)
)
```