Problem 1456: Maximum Number of Vowels in a Substring of Given Length

Problem Information

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

Problem Description

Given a string

s

and an integer

k

, return

the maximum number of vowel letters in any substring of

s

with length

k

.

Vowel letters

in English are

'a'

,

'e'

,

'i'

,

'o'

, and

'u'

.

Example 1:

Input:

s = "abciiidef", k = 3

Output:

3

Explanation:

The substring "iii" contains 3 vowel letters.

Example 2:

Input:

s = "aeiou", k = 2

Output:

2

Explanation:

Any substring of length 2 contains 2 vowels.

Example 3:

Input:

s = "leetcode", k = 3

Output:

2

Explanation:

"lee", "eet" and "ode" contain 2 vowels.

Constraints:

1 <= s.length <= 10

5

s

consists of lowercase English letters.

1 <= k <= s.length

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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

Python:

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

JavaScript:

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

};
```

TypeScript:

```
function maxVowels(s: string, k: number): number {

};
```

C#:

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

}
}
```

C:

```
int maxVowels(char* s, int k) {

}
```

Go:

```
func maxVowels(s string, k int) int {

}
```

Kotlin:

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

}
}
```

Swift:

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

}
}
```

Rust:

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

}
}
```

Ruby:

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

end
```

PHP:

```
class Solution {

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

}
}
```

Dart:

```
class Solution {
int maxVowels(String s, int k) {

}
}
```

Scala:

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

}
```

```
}
```

Elixir:

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

end
end
```

Erlang:

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

Racket:

```
(define/contract (max-vowels s k)
(-> string? exact-integer? exact-integer?)
)
```

Solutions

C++ Solution:

```
/*
* Problem: Maximum Number of Vowels in a Substring of Given Length
* Difficulty: Medium
* Tags: array, string, tree
*
* 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:
int maxVowels(string s, int k) {
```

```
}
};
```

Java Solution:

```
/**
* Problem: Maximum Number of Vowels in a Substring of Given Length
* Difficulty: Medium
* Tags: array, string, tree
*
* 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 int maxVowels(String s, int k) {

}
}
```

Python3 Solution:

```
"""
Problem: Maximum Number of Vowels in a Substring of Given Length
Difficulty: Medium
Tags: array, string, tree

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

Python Solution:

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

JavaScript Solution:

```
/**
* Problem: Maximum Number of Vowels in a Substring of Given Length
* Difficulty: Medium
* Tags: array, string, tree
*
* 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 {number}
*/
var maxVowels = function(s, k) {

};
```

TypeScript Solution:

```
/**
* Problem: Maximum Number of Vowels in a Substring of Given Length
* Difficulty: Medium
* Tags: array, string, tree
*
* 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 maxVowels(s: string, k: number): number {
```

```
};
```

C# Solution:

```
/*
* Problem: Maximum Number of Vowels in a Substring of Given Length
* Difficulty: Medium
* Tags: array, string, tree
*
* 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 int MaxVowels(string s, int k) {

}
}
```

C Solution:

```
/*
* Problem: Maximum Number of Vowels in a Substring of Given Length
* Difficulty: Medium
* Tags: array, string, tree
*
* 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
*/

int maxVowels(char* s, int k) {

}
```

Go Solution:

```
// Problem: Maximum Number of Vowels in a Substring of Given Length
// Difficulty: Medium
```

```
// Tags: array, string, tree
//
// 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 maxVowels(s string, k int) int {

}
```

Kotlin Solution:

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

}
}
```

Swift Solution:

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

}
}
```

Rust Solution:

```
// Problem: Maximum Number of Vowels in a Substring of Given Length
// Difficulty: Medium
// Tags: array, string, tree
//
// 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 max_vowels(s: String, k: i32) -> i32 {

}
}
```

Ruby Solution:

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

end
```

PHP Solution:

```
class Solution {

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

}
}
```

Dart Solution:

```
class Solution {
int maxVowels(String s, int k) {

}
}
```

Scala Solution:

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

}
}
```

Elixir Solution:

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

end
end
```

Erlang Solution:

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

Racket Solution:

```
(define/contract (max-vowels s k)
(-> string? exact-integer? exact-integer?)
)
```