Problem 1189: Maximum Number of Balloons

Problem Information

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

Problem Description

Given a string

text

, you want to use the characters of

text

to form as many instances of the word

"balloon"

as possible.

You can use each character in

text

at most once

. Return the maximum number of instances that can be formed.

Example 1:

Input:

text = "nlaebolko"

Output:

1

Example 2:

loonbalxballpoon

Input:

text = "loonbalxballpoon"

Output:

2

Example 3:

Input:

text = "leetcode"

Output:

0

Constraints:

1 <= text.length <= 10

4

text

consists of lower case English letters only.

Note:

This question is the same as

2287: Rearrange Characters to Make Target String.

Code Snippets

C++:

```
class Solution {
public:
int maxNumberOfBalloons(string text) {

}
};
```

Java:

```
class Solution {
public int maxNumberOfBalloons(String text) {

}
}
```

Python3:

```
class Solution:
def maxNumberOfBalloons(self, text: str) -> int:
```

Python:

```
class Solution(object):
def maxNumberOfBalloons(self, text):
```

```
"""
:type text: str
:rtype: int
"""
```

JavaScript:

```
/**
* @param {string} text
* @return {number}
*/
var maxNumberOfBalloons = function(text) {

};
```

TypeScript:

```
function maxNumberOfBalloons(text: string): number {

};
```

C#:

```
public class Solution {
public int MaxNumberOfBalloons(string text) {

}
}
```

C:

```
int maxNumberOfBalloons(char* text) {

}
```

Go:

```
func maxNumberOfBalloons(text string) int {

}
```

Kotlin:

```
class Solution {
fun maxNumberOfBalloons(text: String): Int {

}
}
```

Swift:

```
class Solution {
func maxNumberOfBalloons(_ text: String) -> Int {

}
}
```

Rust:

```
impl Solution {
pub fn max_number_of_balloons(text: String) -> i32 {

}
}
```

Ruby:

```
# @param {String} text
# @return {Integer}
def max_number_of_balloons(text)

end
```

PHP:

```
class Solution {

/**
* @param String $text
* @return Integer
*/
function maxNumberOfBalloons($text) {

}
}
```

Dart:

```
class Solution {
int maxNumberOfBalloons(String text) {

}
}
```

Scala:

```
object Solution {
def maxNumberOfBalloons(text: String): Int = {

}
}
```

Elixir:

```
defmodule Solution do
@spec max_number_of_balloons(text :: String.t) :: integer
def max_number_of_balloons(text) do

end
end
```

Erlang:

```
-spec max_number_of_balloons(Text :: unicode:unicode_binary()) -> integer().
max_number_of_balloons(Text) ->
.
```

Racket:

```
(define/contract (max-number-of-balloons text)
(-> string? exact-integer?)
)
```

Solutions

C++ Solution:

```
/*
* Problem: Maximum Number of Balloons
* Difficulty: Easy
* Tags: string, hash
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(n) for hash map
*/

class Solution {
public:
int maxNumberOfBalloons(string text) {

}
};
```

Java Solution:

```
/**
* Problem: Maximum Number of Balloons
* Difficulty: Easy
* Tags: string, hash
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(n) for hash map
*/

class Solution {
public int maxNumberOfBalloons(String text) {

}
}
```

Python3 Solution:

```
"""
Problem: Maximum Number of Balloons
Difficulty: Easy
Tags: string, hash
```

```
Approach: String manipulation with hash map or two pointers
Time Complexity: O(n) or O(n log n)
Space Complexity: O(n) for hash map
"""

class Solution:
def maxNumberOfBalloons(self, text: str) -> int:
# TODO: Implement optimized solution
pass
```

Python Solution:

```
class Solution(object):
def maxNumberOfBalloons(self, text):
"""
:type text: str
:rtype: int
"""
```

JavaScript Solution:

```
/**
* Problem: Maximum Number of Balloons
* Difficulty: Easy
* Tags: string, hash
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(n) for hash map
*/

/**
* @param {string} text
* @return {number}
*/
var maxNumberOfBalloons = function(text) {

};
```

TypeScript Solution:

```
/**
* Problem: Maximum Number of Balloons
* Difficulty: Easy
* Tags: string, hash
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(n) for hash map
*/

function maxNumberOfBalloons(text: string): number {

};
```

C# Solution:

```
/*
* Problem: Maximum Number of Balloons
* Difficulty: Easy
* Tags: string, hash
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(n) for hash map
*/

public class Solution {
public int MaxNumberOfBalloons(string text) {

}
}
```

C Solution:

```
/*
* Problem: Maximum Number of Balloons
* Difficulty: Easy
* Tags: string, hash
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(n) for hash map
```

```
*/

int maxNumberOfBalloons(char* text) {

}
```

Go Solution:

```
// Problem: Maximum Number of Balloons
// Difficulty: Easy
// Tags: string, hash
//
// Approach: String manipulation with hash map or two pointers
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(n) for hash map

func maxNumberOfBalloons(text string) int {

}
```

Kotlin Solution:

```
class Solution {
fun maxNumberOfBalloons(text: String): Int {

}
}
```

Swift Solution:

```
class Solution {
func maxNumberOfBalloons(_ text: String) -> Int {

}
}
```

Rust Solution:

```
// Problem: Maximum Number of Balloons
// Difficulty: Easy
// Tags: string, hash
```

```
//
// Approach: String manipulation with hash map or two pointers
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(n) for hash map

impl Solution {
pub fn max_number_of_balloons(text: String) -> i32 {

}
}
```

Ruby Solution:

```
# @param {String} text
# @return {Integer}
def max_number_of_balloons(text)

end
```

PHP Solution:

```
class Solution {

/**
* @param String $text
* @return Integer
*/
function maxNumberOfBalloons($text) {

}
}
```

Dart Solution:

```
class Solution {
int maxNumberOfBalloons(String text) {

}
}
```

Scala Solution:

```
object Solution {
def maxNumberOfBalloons(text: String): Int = {

}
}
```

Elixir Solution:

```
defmodule Solution do
@spec max_number_of_balloons(text :: String.t) :: integer
def max_number_of_balloons(text) do

end
end
```

Erlang Solution:

```
-spec max_number_of_balloons(Text :: unicode:unicode_binary()) -> integer().
max_number_of_balloons(Text) ->
.
```

Racket Solution:

```
(define/contract (max-number-of-balloons text)
(-> string? exact-integer?)
)
```