

Problem 3081: Replace Question Marks in String to Minimize Its Value

Problem Information

Difficulty: Medium

Acceptance Rate: 0.00%

Paid Only: No

Problem Description

You are given a string

s

s[i]

is either a lowercase English letter or

'?'

For a string

t

having length

m

containing

only

lowercase English letters, we define the function

$\text{cost}(i)$

for an index

i

as the number of characters

equal

to

$t[i]$

that appeared before it, i.e. in the range

$[0, i - 1]$

.

The

value

of

t

is the

sum

of

$\text{cost}(i)$

for all indices

i

.

For example, for the string

$t = "aab"$

:

$\text{cost}(0) = 0$

$\text{cost}(1) = 1$

$\text{cost}(2) = 0$

Hence, the value of

"aab"

is

$0 + 1 + 0 = 1$

.

Your task is to

replace all

occurrences of

'?'

in

s

with any lowercase English letter so that the

value

of

s

is

minimized

.

Return

a string denoting the modified string with replaced occurrences of

'?'

. If there are multiple strings resulting in the

minimum value

, return the

lexicographically smallest

one.

Example 1:

Input:

`s = "???"`

Output:

"abc"

Explanation:

In this example, we can replace the occurrences of

'?'

to make

s

equal to

"abc"

.

For

"abc"

,

$\text{cost}(0) = 0$

,

$\text{cost}(1) = 0$

, and

$\text{cost}(2) = 0$

.

The value of

"abc"

is

0

.

Some other modifications of

s

that have a value of

0

are

"cba"

,

"abz"

, and,

"hey"

.

Among all of them, we choose the lexicographically smallest.

Example 2:

Input:

s = "a?a?"

Output:

"abac"

Explanation:

In this example, the occurrences of

'?'

can be replaced to make

s

equal to

"abac"

.

For

"abac"

,

$\text{cost}(0) = 0$

,

$\text{cost}(1) = 0$

,

$\text{cost}(2) = 1$

, and

$\text{cost}(3) = 0$

.

The value of

"abac"

is

1

.

Constraints:

$1 \leq s.length \leq 10$

5

$s[i]$

is either a lowercase English letter or

'?'

.

Code Snippets

C++:

```
class Solution {
public:
    string minimizeStringValue(string s) {
        }
};
```

Java:

```
class Solution {
    public String minimizeStringValue(String s) {
```

```
}
```

```
}
```

Python3:

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

Python:

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

JavaScript:

```
/**  
 * @param {string} s  
 * @return {string}  
 */  
var minimizeStringValue = function(s) {  
  
};
```

TypeScript:

```
function minimizeStringValue(s: string): string {  
  
};
```

C#:

```
public class Solution {  
    public string MinimizeStringValue(string s) {  
  
    }  
}
```

C:

```
char* minimizeStringValue(char* s) {  
  
}
```

Go:

```
func minimizeStringValue(s string) string {  
  
}
```

Kotlin:

```
class Solution {  
    fun minimizeStringValue(s: String): String {  
  
    }  
}
```

Swift:

```
class Solution {  
    func minimizeStringValue(_ s: String) -> String {  
  
    }  
}
```

Rust:

```
impl Solution {  
    pub fn minimize_string_value(s: String) -> String {  
  
    }  
}
```

Ruby:

```
# @param {String} s  
# @return {String}  
def minimize_string_value(s)  
  
end
```

PHP:

```
class Solution {  
  
    /**  
     * @param String $s  
     * @return String  
     */  
    function minimizeStringValue($s) {  
  
    }  
}
```

Dart:

```
class Solution {  
    String minimizeStringValue(String s) {  
  
    }  
}
```

Scala:

```
object Solution {  
    def minimizeStringValue(s: String): String = {  
  
    }  
}
```

Elixir:

```
defmodule Solution do  
  @spec minimize_string_value(s :: String.t) :: String.t  
  def minimize_string_value(s) do  
  
  end  
end
```

Erlang:

```
-spec minimize_string_value(S :: unicode:unicode_binary()) ->  
  unicode:unicode_binary().  
minimize_string_value(S) ->
```

.

Racket:

```
(define/contract (minimize-string-value s)
  (-> string? string?))
```

Solutions

C++ Solution:

```
/*
 * Problem: Replace Question Marks in String to Minimize Its Value
 * Difficulty: Medium
 * Tags: string, graph, greedy, hash, sort, queue, heap
 *
 * 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:
    string minimizeStringValue(string s) {

    }
};
```

Java Solution:

```
/**
 * Problem: Replace Question Marks in String to Minimize Its Value
 * Difficulty: Medium
 * Tags: string, graph, greedy, hash, sort, queue, heap
 *
 * 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 String minimizeStringValue(String s) {  
  
    }  
}
```

Python3 Solution:

```
"""  
  
Problem: Replace Question Marks in String to Minimize Its Value  
Difficulty: Medium  
Tags: string, graph, greedy, hash, sort, queue, heap  
  
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 minimizeStringValue(self, s: str) -> str:  
        # TODO: Implement optimized solution  
        pass
```

Python Solution:

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

JavaScript Solution:

```
/**  
 * Problem: Replace Question Marks in String to Minimize Its Value  
 * Difficulty: Medium  
 * Tags: string, graph, greedy, hash, sort, queue, heap  
 * 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} s
 * @return {string}
 */
var minimizeStringValue = function(s) {

};

```

TypeScript Solution:

```

/** 
 * Problem: Replace Question Marks in String to Minimize Its Value
 * Difficulty: Medium
 * Tags: string, graph, greedy, hash, sort, queue, heap
 *
 * 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 minimizeStringValue(s: string): string {
}

```

C# Solution:

```

/*
 * Problem: Replace Question Marks in String to Minimize Its Value
 * Difficulty: Medium
 * Tags: string, graph, greedy, hash, sort, queue, heap
 *
 * 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 string MinimizeStringValue(string s) {  
    }  
}
```

C Solution:

```
/*  
 * Problem: Replace Question Marks in String to Minimize Its Value  
 * Difficulty: Medium  
 * Tags: string, graph, greedy, hash, sort, queue, heap  
 *  
 * 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  
 */  
  
char* minimizeStringValue(char* s) {  
}
```

Go Solution:

```
// Problem: Replace Question Marks in String to Minimize Its Value  
// Difficulty: Medium  
// Tags: string, graph, greedy, hash, sort, queue, heap  
//  
// 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 minimizeStringValue(s string) string {  
}
```

Kotlin Solution:

```
class Solution {  
    fun minimizeStringValue(s: String): String {  
    }
```

```
}
```

Swift Solution:

```
class Solution {  
func minimizeStringValue(_ s: String) -> String {  
  
}  
}
```

Rust Solution:

```
// Problem: Replace Question Marks in String to Minimize Its Value  
// Difficulty: Medium  
// Tags: string, graph, greedy, hash, sort, queue, heap  
//  
// 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 minimize_string_value(s: String) -> String {  
  
}  
}
```

Ruby Solution:

```
# @param {String} s  
# @return {String}  
def minimize_string_value(s)  
  
end
```

PHP Solution:

```
class Solution {  
  
/**  
* @param String $s  
* @return String
```

```
*/  
function minimizeStringValue($s) {  
  
}  
}  
}
```

Dart Solution:

```
class Solution {  
String minimizeStringValue(String s) {  
  
}  
}  
}
```

Scala Solution:

```
object Solution {  
def minimizeStringValue(s: String): String = {  
  
}  
}
```

Elixir Solution:

```
defmodule Solution do  
@spec minimize_string_value(s :: String.t) :: String.t  
def minimize_string_value(s) do  
  
end  
end
```

Erlang Solution:

```
-spec minimize_string_value(S :: unicode:unicode_binary()) ->  
unicode:unicode_binary().  
minimize_string_value(S) ->  
.
```

Racket Solution:

```
(define/contract (minimize-string-value s)
(-> string? string?))
)
```