

Problem 1736: Latest Time by Replacing Hidden Digits

Problem Information

Difficulty: **Easy**

Acceptance Rate: 0.00%

Paid Only: No

Problem Description

You are given a string

time

in the form of

hh:mm

, where some of the digits in the string are hidden (represented by

?).

The valid times are those inclusively between

00:00

and

23:59

[Return](#)

the latest valid time you can get from

time

by replacing the hidden

digits

.

Example 1:

Input:

time = "2?:?0"

Output:

"23:50"

Explanation:

The latest hour beginning with the digit '2' is 23 and the latest minute ending with the digit '0' is 50.

Example 2:

Input:

time = "0?:3?"

Output:

"09:39"

Example 3:

Input:

```
time = "1?22"
```

Output:

```
"19:22"
```

Constraints:

time

is in the format

hh:mm

.

It is guaranteed that you can produce a valid time from the given string.

Code Snippets

C++:

```
class Solution {
public:
    string maximumTime(string time) {
        }
    };
}
```

Java:

```
class Solution {
    public String maximumTime(String time) {
        }
    }
}
```

Python3:

```
class Solution:  
    def maximumTime(self, time: str) -> str:
```

Python:

```
class Solution(object):  
    def maximumTime(self, time):  
        """  
        :type time: str  
        :rtype: str  
        """
```

JavaScript:

```
/**  
 * @param {string} time  
 * @return {string}  
 */  
var maximumTime = function(time) {  
  
};
```

TypeScript:

```
function maximumTime(time: string): string {  
  
};
```

C#:

```
public class Solution {  
    public string MaximumTime(string time) {  
  
    }  
}
```

C:

```
char* maximumTime(char* time) {  
  
}
```

Go:

```
func maximumTime(time string) string {  
}  
}
```

Kotlin:

```
class Solution {  
    fun maximumTime(time: String): String {  
        return ""  
    }  
}
```

Swift:

```
class Solution {  
    func maximumTime(_ time: String) -> String {  
        return ""  
    }  
}
```

Rust:

```
impl Solution {  
    pub fn maximum_time(time: String) -> String {  
        return ""  
    }  
}
```

Ruby:

```
# @param {String} time  
# @return {String}  
def maximum_time(time)  
  
end
```

PHP:

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

```
*/  
function maximumTime($time) {  
  
}  
}  
}
```

Dart:

```
class Solution {  
String maximumTime(String time) {  
  
}  
}  
}
```

Scala:

```
object Solution {  
def maximumTime(time: String): String = {  
  
}  
}
```

Elixir:

```
defmodule Solution do  
@spec maximum_time(String.t) :: String.t  
def maximum_time(time) do  
  
end  
end
```

Erlang:

```
-spec maximum_time(unicode:unicode_binary()) ->  
unicode:unicode_binary().  
maximum_time(Time) ->  
.
```

Racket:

```
(define/contract (maximum-time time)  
(-> string? string?))
```

```
)
```

Solutions

C++ Solution:

```
/*
 * Problem: Latest Time by Replacing Hidden Digits
 * Difficulty: Easy
 * Tags: string, greedy
 *
 * Approach: String manipulation with hash map or two pointers
 * Time Complexity: O(n) or O(n log n)
 * Space Complexity: O(1) to O(n) depending on approach
 */

class Solution {
public:
    string maximumTime(string time) {

    }
};
```

Java Solution:

```
/**
 * Problem: Latest Time by Replacing Hidden Digits
 * Difficulty: Easy
 * Tags: string, greedy
 *
 * Approach: String manipulation with hash map or two pointers
 * Time Complexity: O(n) or O(n log n)
 * Space Complexity: O(1) to O(n) depending on approach
 */

class Solution {
    public String maximumTime(String time) {

    }
}
```

Python3 Solution:

```
"""
Problem: Latest Time by Replacing Hidden Digits
Difficulty: Easy
Tags: string, greedy

Approach: String manipulation with hash map or two pointers
Time Complexity: O(n) or O(n log n)
Space Complexity: O(1) to O(n) depending on approach
"""

class Solution:

    def maximumTime(self, time: str) -> str:
        # TODO: Implement optimized solution
        pass
```

Python Solution:

```
class Solution(object):

    def maximumTime(self, time):

        """
:type time: str
:rtype: str
"""


```

JavaScript Solution:

```
/**
 * Problem: Latest Time by Replacing Hidden Digits
 * Difficulty: Easy
 * Tags: string, greedy
 *
 * Approach: String manipulation with hash map or two pointers
 * Time Complexity: O(n) or O(n log n)
 * Space Complexity: O(1) to O(n) depending on approach
 */

/**
 * @param {string} time
 * @return {string}
 */
```

```
var maximumTime = function(time) {  
};
```

TypeScript Solution:

```
/**  
 * Problem: Latest Time by Replacing Hidden Digits  
 * Difficulty: Easy  
 * Tags: string, greedy  
 *  
 * Approach: String manipulation with hash map or two pointers  
 * Time Complexity: O(n) or O(n log n)  
 * Space Complexity: O(1) to O(n) depending on approach  
 */  
  
function maximumTime(time: string): string {  
};
```

C# Solution:

```
/*  
 * Problem: Latest Time by Replacing Hidden Digits  
 * Difficulty: Easy  
 * Tags: string, greedy  
 *  
 * Approach: String manipulation with hash map or two pointers  
 * Time Complexity: O(n) or O(n log n)  
 * Space Complexity: O(1) to O(n) depending on approach  
 */  
  
public class Solution {  
    public string MaximumTime(string time) {  
        }  
    }  
}
```

C Solution:

```

/*
 * Problem: Latest Time by Replacing Hidden Digits
 * Difficulty: Easy
 * Tags: string, greedy
 *
 * Approach: String manipulation with hash map or two pointers
 * Time Complexity: O(n) or O(n log n)
 * Space Complexity: O(1) to O(n) depending on approach
 */

char* maximumTime(char* time) {

}

```

Go Solution:

```

// Problem: Latest Time by Replacing Hidden Digits
// Difficulty: Easy
// Tags: string, greedy
//
// Approach: String manipulation with hash map or two pointers
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(1) to O(n) depending on approach

func maximumTime(time string) string {

}

```

Kotlin Solution:

```

class Solution {
    fun maximumTime(time: String): String {
        return ""
    }
}

```

Swift Solution:

```

class Solution {
    func maximumTime(_ time: String) -> String {
        return ""
    }
}

```

```
}
```

Rust Solution:

```
// Problem: Latest Time by Replacing Hidden Digits
// Difficulty: Easy
// Tags: string, greedy
//
// Approach: String manipulation with hash map or two pointers
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(1) to O(n) depending on approach

impl Solution {
    pub fn maximum_time(time: String) -> String {
        //
    }
}
```

Ruby Solution:

```
# @param {String} time
# @return {String}
def maximum_time(time)

end
```

PHP Solution:

```
class Solution {

    /**
     * @param String $time
     * @return String
     */
    function maximumTime($time) {

    }
}
```

Dart Solution:

```
class Solution {  
    String maximumTime(String time) {  
        }  
    }  
}
```

Scala Solution:

```
object Solution {  
    def maximumTime(time: String): String = {  
        }  
    }  
}
```

Elixir Solution:

```
defmodule Solution do  
    @spec maximum_time(time :: String.t) :: String.t  
    def maximum_time(time) do  
  
    end  
end
```

Erlang Solution:

```
-spec maximum_time(Time :: unicode:unicode_binary()) ->  
unicode:unicode_binary().  
maximum_time(Time) ->  
.
```

Racket Solution:

```
(define/contract (maximum-time time)  
(-> string? string?)  
)
```