Problem 2264: Largest 3-Same-Digit Number in String

Problem Information

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

Problem Description

You are given a string

num

representing a large integer. An integer is

good

if it meets the following conditions:

It is a

substring

of

num

with length

3

.

It consists of only one unique digit.

Return

the

maximum good

integer as a

string

or an empty string

""

if no such integer exists

.

Note:

A

substring

is a contiguous sequence of characters within a string.

There may be

leading zeroes

in

num

or a good integer.

Example 1:

Input:

num = "6

777

133339"

Output:

"777"

Explanation:

There are two distinct good integers: "777" and "333". "777" is the largest, so we return "777".

Example 2:

Input:

num = "23

000

19"

Output:

"000"

Explanation:

"000" is the only good integer.

Example 3:

Input:

num = "42352338"

Output:

""

Explanation:

No substring of length 3 consists of only one unique digit. Therefore, there are no good integers.

Constraints:

3 <= num.length <= 1000

num

only consists of digits.

Code Snippets

C++:

```
class Solution {
public:
string largestGoodInteger(string num) {

}
};
```

Java:

```
class Solution {
public String largestGoodInteger(String num) {

}
}
```

Python3:

```
class Solution:
def largestGoodInteger(self, num: str) -> str:
```

Python:

```
class Solution(object):
def largestGoodInteger(self, num):
"""
:type num: str
:rtype: str
"""
```

JavaScript:

```
/**
* @param {string} num
* @return {string}
*/
var largestGoodInteger = function(num) {

};
```

TypeScript:

```
function largestGoodInteger(num: string): string {

};
```

C#:

```
public class Solution {
public string LargestGoodInteger(string num) {

}
}
```

C:

```
char* largestGoodInteger(char* num) {

}
```

Go:

```
func largestGoodInteger(num string) string {

}
```

Kotlin:

```
class Solution {
fun largestGoodInteger(num: String): String {

}
}
```

Swift:

```
class Solution {
func largestGoodInteger(_ num: String) -> String {

}
}
```

Rust:

```
impl Solution {
pub fn largest_good_integer(num: String) -> String {

}
}
```

Ruby:

```
# @param {String} num
# @return {String}
def largest_good_integer(num)

end
```

PHP:

```
class Solution {

/**
* @param String $num
* @return String
*/
function largestGoodInteger($num) {

}
}
```

Dart:

```
class Solution {
String largestGoodInteger(String num) {

}
}
```

Scala:

```
object Solution {
def largestGoodInteger(num: String): String = {

}
}
```

Elixir:

```
defmodule Solution do
@spec largest_good_integer(num :: String.t) :: String.t
def largest_good_integer(num) do

end
end
```

Erlang:

```
-spec largest_good_integer(Num :: unicode:unicode_binary()) ->
unicode:unicode_binary().
largest_good_integer(Num) ->
.
```

Racket:

```
(define/contract (largest-good-integer num)
(-> string? string?)
)
```

Solutions

C++ Solution:

```
/*
* Problem: Largest 3-Same-Digit Number in String
* Difficulty: Easy
* Tags: string, tree
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

class Solution {
public:
string largestGoodInteger(string num) {

}
};
```

Java Solution:

```
/**
* Problem: Largest 3-Same-Digit Number in String
* Difficulty: Easy
* Tags: string, tree
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

class Solution {
public String largestGoodInteger(String num) {

}
}
```

Python3 Solution:

```
"""
Problem: Largest 3-Same-Digit Number in String
Difficulty: Easy
Tags: string, tree

Approach: String manipulation with hash map or two pointers
Time Complexity: O(n) or O(n log n)
Space Complexity: O(h) for recursion stack where h is height
"""

class Solution:
def largestGoodInteger(self, num: str) -> str:
# TODO: Implement optimized solution
pass
```

Python Solution:

```
class Solution(object):
def largestGoodInteger(self, num):
"""
:type num: str
:rtype: str
"""
```

JavaScript Solution:

```
/**
* Problem: Largest 3-Same-Digit Number in String
* Difficulty: Easy
* Tags: string, tree
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

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

```
var largestGoodInteger = function(num) {

};
```

TypeScript Solution:

```
/**
* Problem: Largest 3-Same-Digit Number in String
* Difficulty: Easy
* Tags: string, tree
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

function largestGoodInteger(num: string): string {

};
```

C# Solution:

```
/*
* Problem: Largest 3-Same-Digit Number in String
* Difficulty: Easy
* Tags: string, tree
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

public class Solution {
public string LargestGoodInteger(string num) {

}
}
```

C Solution:

```
/*
* Problem: Largest 3-Same-Digit Number in String
* Difficulty: Easy
* Tags: string, tree
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

char* largestGoodInteger(char* num) {

}
```

Go Solution:

```
// Problem: Largest 3-Same-Digit Number in String
// Difficulty: Easy
// Tags: string, tree
//
// Approach: String manipulation with hash map or two pointers
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(h) for recursion stack where h is height

func largestGoodInteger(num string) string {

}
```

Kotlin Solution:

```
class Solution {
fun largestGoodInteger(num: String): String {

}
}
```

Swift Solution:

```
class Solution {
func largestGoodInteger(_ num: String) -> String {

}

}
```

Rust Solution:

```
// Problem: Largest 3-Same-Digit Number in String
// Difficulty: Easy
// Tags: string, tree
//
// Approach: String manipulation with hash map or two pointers
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(h) for recursion stack where h is height

impl Solution {
pub fn largest_good_integer(num: String) -> String {

}
}
```

Ruby Solution:

```
# @param {String} num
# @return {String}
def largest_good_integer(num)

end
```

PHP Solution:

```
class Solution {

/**
* @param String $num
* @return String
*/
function largestGoodInteger($num) {

}
}
```

Dart Solution:

```
class Solution {
String largestGoodInteger(String num) {

}
}
```

Scala Solution:

```
object Solution {
def largestGoodInteger(num: String): String = {

}
}
```

Elixir Solution:

```
defmodule Solution do
@spec largest_good_integer(num :: String.t) :: String.t
def largest_good_integer(num) do

end
end
```

Erlang Solution:

```
-spec largest_good_integer(Num :: unicode:unicode_binary()) ->
unicode:unicode_binary().
largest_good_integer(Num) ->
.
```

Racket Solution:

```
(define/contract (largest-good-integer num)
(-> string? string?)
)
```