Problem 400: Nth Digit

Problem Information

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

Problem Description

Given an integer

n

, return the

n

th

digit of the infinite integer sequence

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, ...]

.

Example 1:

Input:

n = 3

Output:

3

Example 2:

Input:

n = 11

Output:

0

Explanation:

The 11

th

digit of the sequence 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, ... is a 0, which is part of the number 10.

Constraints:

1 <= n <= 2

31

- 1

Code Snippets

C++:

```
class Solution {
public:
int findNthDigit(int n) {

}
};
```

Java:

```
class Solution {
public int findNthDigit(int n) {

}
}
```

Python3:

```
class Solution:
def findNthDigit(self, n: int) -> int:
```

Python:

```
class Solution(object):
def findNthDigit(self, n):
"""
:type n: int
:rtype: int
"""
```

JavaScript:

```
/**
* @param {number} n
* @return {number}
*/
var findNthDigit = function(n) {

};
```

TypeScript:

```
function findNthDigit(n: number): number {

};
```

C#:

```
public class Solution {
public int FindNthDigit(int n) {

}
}
```

C:

```
int findNthDigit(int n) {

}
```

Go:

```
func findNthDigit(n int) int {

}
```

Kotlin:

```
class Solution {
fun findNthDigit(n: Int): Int {

}
}
```

Swift:

```
class Solution {
func findNthDigit(_ n: Int) -> Int {

}
}
```

Rust:

```
impl Solution {
pub fn find_nth_digit(n: i32) -> i32 {

}
}
```

Ruby:

```
# @param {Integer} n
# @return {Integer}
def find_nth_digit(n)

end
```

PHP:

```
class Solution {

/**
* @param Integer $n
* @return Integer
*/
function findNthDigit($n) {

}
}
```

Dart:

```
class Solution {
int findNthDigit(int n) {

}
}
```

Scala:

```
object Solution {
def findNthDigit(n: Int): Int = {

}
}
```

Elixir:

```
defmodule Solution do
@spec find_nth_digit(n :: integer) :: integer
def find_nth_digit(n) do

end
end
```

Erlang:

```
-spec find_nth_digit(N :: integer()) -> integer().
find_nth_digit(N) ->
.
```

Racket:

```
(define/contract (find-nth-digit n)
(-> exact-integer? exact-integer?)
)
```

Solutions

C++ Solution:

```
/*
* Problem: Nth Digit
* Difficulty: Medium
* Tags: math, search
*
* Approach: Optimized algorithm based on problem constraints
* Time Complexity: O(n) to O(n^2) depending on approach
* Space Complexity: O(1) to O(n) depending on approach
*/

class Solution {
public:
int findNthDigit(int n) {

}
};
```

Java Solution:

```
/**
* Problem: Nth Digit
* Difficulty: Medium
* Tags: math, search
*
* Approach: Optimized algorithm based on problem constraints
* Time Complexity: O(n) to O(n^2) depending on approach
* Space Complexity: O(1) to O(n) depending on approach
*/

class Solution {
public int findNthDigit(int n) {
}
}
```

Python3 Solution:

```
"""
Problem: Nth Digit
Difficulty: Medium
Tags: math, search

Approach: Optimized algorithm based on problem constraints
Time Complexity: O(n) to O(n^2) depending on approach
Space Complexity: O(1) to O(n) depending on approach
"""

class Solution:
def findNthDigit(self, n: int) -> int:
# TODO: Implement optimized solution
pass
```

Python Solution:

```
class Solution(object):
def findNthDigit(self, n):
"""
:type n: int
:rtype: int
"""
```

JavaScript Solution:

```
/**
* Problem: Nth Digit
* Difficulty: Medium
* Tags: math, search
*
* Approach: Optimized algorithm based on problem constraints
* Time Complexity: O(n) to O(n^2) depending on approach
* Space Complexity: O(1) to O(n) depending on approach
*/
```

```
/**
* @param {number} n
* @return {number}
*/
var findNthDigit = function(n) {

};
```

TypeScript Solution:

```
/**
* Problem: Nth Digit
* Difficulty: Medium
* Tags: math, search
*
* Approach: Optimized algorithm based on problem constraints
* Time Complexity: O(n) to O(n^2) depending on approach
* Space Complexity: O(1) to O(n) depending on approach
*/

function findNthDigit(n: number): number {

};
```

C# Solution:

```
/*
* Problem: Nth Digit
* Difficulty: Medium
* Tags: math, search
*
* Approach: Optimized algorithm based on problem constraints
* Time Complexity: O(n) to O(n^2) depending on approach
* Space Complexity: O(1) to O(n) depending on approach
*/

public class Solution {
public int FindNthDigit(int n) {

}

}
```

C Solution:

```
/*
* Problem: Nth Digit
* Difficulty: Medium
* Tags: math, search
*
* Approach: Optimized algorithm based on problem constraints
* Time Complexity: O(n) to O(n^2) depending on approach
* Space Complexity: O(1) to O(n) depending on approach
*/

int findNthDigit(int n) {

}
```

Go Solution:

```
// Problem: Nth Digit
// Difficulty: Medium
// Tags: math, search
//
// Approach: Optimized algorithm based on problem constraints
// Time Complexity: O(n) to O(n^2) depending on approach
// Space Complexity: O(1) to O(n) depending on approach

func findNthDigit(n int) int {

}
```

Kotlin Solution:

```
class Solution {
fun findNthDigit(n: Int): Int {

}
}
```

Swift Solution:

```
class Solution {
func findNthDigit(_ n: Int) -> Int {

}
}
```

Rust Solution:

```
// Problem: Nth Digit
// Difficulty: Medium
// Tags: math, search
//
// Approach: Optimized algorithm based on problem constraints
// Time Complexity: O(n) to O(n^2) depending on approach
// Space Complexity: O(1) to O(n) depending on approach

impl Solution {
pub fn find_nth_digit(n: i32) -> i32 {

}
}
```

Ruby Solution:

```
# @param {Integer} n
# @return {Integer}
def find_nth_digit(n)

end
```

PHP Solution:

```
class Solution {

/**
* @param Integer $n
* @return Integer
*/
function findNthDigit($n) {

}
}
```

Dart Solution:

```
class Solution {
int findNthDigit(int n) {

}
}
```

Scala Solution:

```
object Solution {
def findNthDigit(n: Int): Int = {

}
}
```

Elixir Solution:

```
defmodule Solution do
@spec find_nth_digit(n :: integer) :: integer
def find_nth_digit(n) do

end
end
```

Erlang Solution:

```
-spec find_nth_digit(N :: integer()) -> integer().
find_nth_digit(N) ->
.
```

Racket Solution:

```
(define/contract (find-nth-digit n)
(-> exact-integer? exact-integer?)
)
```