Problem 470: Implement Rand10() Using Rand7()

Problem Information

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

Problem Description

Given the

API

rand7()

that generates a uniform random integer in the range

[1, 7]

, write a function

rand10()

that generates a uniform random integer in the range

[1, 10]

. You can only call the API

rand7()

, and you shouldn't call any other API. Please

do not

use a language's built-in random API.

Each test case will have one

internal

argument

n

, the number of times that your implemented function

rand10()

will be called while testing. Note that this is

not an argument

passed to

rand10()

.

Example 1:

Input:

n = 1

Output:

[2]

Example 2:

Input:

n = 2

Output:

[2,8]

Example 3:

Input:

n = 3

Output:

[3,8,10]

Constraints:

1 <= n <= 10

5

Follow up:

What is the

expected value

for the number of calls to

rand7()

function?

Could you minimize the number of calls to

rand7()

?

Code Snippets

C++:

```
// The rand7() API is already defined for you.
// int rand7();
// @return a random integer in the range 1 to 7

class Solution {
public:
int rand10() {

}
};
```

Java:

```
/**
* The rand7() API is already defined in the parent class SolBase.
* public int rand7();
* @return a random integer in the range 1 to 7
*/
class Solution extends SolBase {
public int rand10() {

}
}
```

Python3:

```
# The rand7() API is already defined for you.
# def rand7():
# @return a random integer in the range 1 to 7

class Solution:
def rand10(self):
"""
:rtype: int
"""
```

Python:

```
# The rand7() API is already defined for you.
# def rand7():
# @return a random integer in the range 1 to 7

class Solution(object):
def rand10(self):
"""
:rtype: int
"""
```

JavaScript:

```
/**
* The rand7() API is already defined for you.
* var rand7 = function() {}
* @return {number} a random integer in the range 1 to 7
*/
var rand10 = function() {

};
```

TypeScript:

```
/**
* The rand7() API is already defined for you.
* function rand7(): number {}
* @return a random integer in the range 1 to 7
*/

function rand10(): number {

};
```

C#:

```
/**
* The Rand7() API is already defined in the parent class SolBase.
* public int Rand7();
* @return a random integer in the range 1 to 7
*/
public class Solution : SolBase {
public int Rand10() {
```

```
}
}
```

C:

```
// The rand7() API is already defined for you.
// int rand7();
// @return a random integer in the range 1 to 7

int rand10() {

}
```

Go:

```
func rand10() int {

}
```

Kotlin:

```
/**
* The rand7() API is already defined in the parent class SolBase.
* fun rand7(): Int {}
* @return a random integer in the range 1 to 7
*/
class Solution : SolBase() {
fun rand10(): Int {

}
}
```

Swift:

```
/**
* The rand7() API is already defined in the parent class SolBase.
* func rand7() -> Int = {}
* @return a random integer in the range 1 to 7
*/
class Solution : SolBase {
func rand10() -> Int {
```

```
}
}
```

Rust:

```
/**
* The rand7() API is already defined for you.
* @return a random integer in the range 1 to 7
* fn rand7() -> i32;
*/

impl Solution {
pub fn rand10() -> i32 {

}
}
```

Ruby:

```
# The rand7() API is already defined for you.
# def rand7()
# @return {Integer} a random integer in the range 1 to 7

def rand10()

end
```

PHP:

```
/*
* The rand7() API is already defined for you.
* @return a random integer in the range 1 to 7
* function rand7();
*/

class Solution {
/**
* @param
* @return Integer
*/
function rand10() {
```

```
}
}
```

Scala:

```
/**
* The rand7() API is already defined in the parent class SolBase.
* def rand7(): Int = {}
* @return a random integer in the range 1 to 7
*/
object Solution extends SolBase {
def rand10(): Int = {

}
}
```

Solutions

C++ Solution:

```
/*
* Problem: Implement Rand10() Using Rand7()
* Difficulty: Medium
* Tags: math
*
* 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
*/

// The rand7() API is already defined for you.
// int rand7();
// @return a random integer in the range 1 to 7

class Solution {
public:
int rand10() {

}
};
```

Java Solution:

```
/**
* Problem: Implement Rand10() Using Rand7()
* Difficulty: Medium
* Tags: math
*
* 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
*/

/**
* The rand7() API is already defined in the parent class SolBase.
* public int rand7();
* @return a random integer in the range 1 to 7
*/
class Solution extends SolBase {
public int rand10() {

}
}
```

Python3 Solution:

```
# The rand7() API is already defined for you.
# def rand7():
# @return a random integer in the range 1 to 7

class Solution:
def rand10(self):
"""
:rtype: int
"""
```

Python Solution:

```
# The rand7() API is already defined for you.
# def rand7():
# @return a random integer in the range 1 to 7

class Solution(object):
```

```
def rand10(self):
"""
:rtype: int
"""
```

JavaScript Solution:

```
/**
* Problem: Implement Rand10() Using Rand7()
* Difficulty: Medium
* Tags: math
*
* 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
*/

/**
* The rand7() API is already defined for you.
* var rand7 = function() {}
* @return {number} a random integer in the range 1 to 7
*/
var rand10 = function() {

};
```

TypeScript Solution:

```
/**
* Problem: Implement Rand10() Using Rand7()
* Difficulty: Medium
* Tags: math
*
* 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
*/

/**
* The rand7() API is already defined for you.
* function rand7(): number {}
```

```
* @return a random integer in the range 1 to 7
*/

function rand10(): number {

};
```

C# Solution:

```
/*
* Problem: Implement Rand10() Using Rand7()
* Difficulty: Medium
* Tags: math
*
* 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
*/

/**
* The Rand7() API is already defined in the parent class SolBase.
* public int Rand7();
* @return a random integer in the range 1 to 7
*/
public class Solution : SolBase {
public int Rand10() {

}
}
```

C Solution:

```
/*
* Problem: Implement Rand10() Using Rand7()
* Difficulty: Medium
* Tags: math
*
* 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
*/
```

```
// The rand7() API is already defined for you.
// int rand7();
// @return a random integer in the range 1 to 7

int rand10() {

}
```

Go Solution:

```
// Problem: Implement Rand10() Using Rand7()
// Difficulty: Medium
// Tags: math
//
// 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 rand10() int {

}
```

Kotlin Solution:

```
/**
* The rand7() API is already defined in the parent class SolBase.
* fun rand7(): Int {}
* @return a random integer in the range 1 to 7
*/
class Solution : SolBase() {
fun rand10(): Int {

}
}
```

Swift Solution:

```
/**
* The rand7() API is already defined in the parent class SolBase.
* func rand7() -> Int = {}
```

```
* @return a random integer in the range 1 to 7
*/
class Solution : SolBase {
func rand10() -> Int {

}
}
```

Rust Solution:

```
// Problem: Implement Rand10() Using Rand7()
// Difficulty: Medium
// Tags: math
//
// 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

/**
* The rand7() API is already defined for you.
* @return a random integer in the range 1 to 7
* fn rand7() -> i32;
*/

impl Solution {
pub fn rand10() -> i32 {

}
}
```

Ruby Solution:

```
# The rand7() API is already defined for you.
# def rand7()
# @return {Integer} a random integer in the range 1 to 7

def rand10()

end
```

PHP Solution:

```
/*
* The rand7() API is already defined for you.
* @return a random integer in the range 1 to 7
* function rand7();
*/

class Solution {
/**
* @param
* @return Integer
*/
function rand10() {

}
}
```

Scala Solution:

```
/**
* The rand7() API is already defined in the parent class SolBase.
* def rand7(): Int = {}
* @return a random integer in the range 1 to 7
*/
object Solution extends SolBase {
def rand10(): Int = {

}
}
```