Problem 397: Integer Replacement

Problem Information

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

Problem Description

Given a positive integer

n

, you can apply one of the following operations:

If

n

is even, replace

n

with

n / 2

.

If

n

is odd, replace

n

with either

n + 1

or

n - 1

.

Return

the minimum number of operations needed for

n

to become

1

.

Example 1:

Input:

n = 8

Output:

3

Explanation:

8 -> 4 -> 2 -> 1

Example 2:

Input:

n = 7

Output:

4

Explanation:

7 -> 8 -> 4 -> 2 -> 1 or 7 -> 6 -> 3 -> 2 -> 1

Example 3:

Input:

n = 4

Output:

2

Constraints:

1 <= n <= 2

31

- 1

Code Snippets

C++:

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

```
}
};
```

Java:

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

}
}
```

Python3:

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

Python:

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

JavaScript:

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

};
```

TypeScript:

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

};
```

C#:

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

}
}
```

C:

```
int integerReplacement(int n) {

}
```

Go:

```
func integerReplacement(n int) int {

}
```

Kotlin:

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

}
}
```

Swift:

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

}
}
```

Rust:

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

}
}
```

Ruby:

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

end
```

PHP:

```
class Solution {

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

}
}
```

Dart:

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

}
}
```

Scala:

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

}
}
```

Elixir:

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

end
end
```

Erlang:

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

Racket:

```
(define/contract (integer-replacement n)
(-> exact-integer? exact-integer?)
)
```

Solutions

C++ Solution:

```
/*
* Problem: Integer Replacement
* Difficulty: Medium
* Tags: dp, greedy
*
* Approach: Dynamic programming with memoization or tabulation
* Time Complexity: O(n * m) where n and m are problem dimensions
* Space Complexity: O(n) or O(n * m) for DP table
*/

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

}
};
```

Java Solution:

```
/**
* Problem: Integer Replacement
* Difficulty: Medium
* Tags: dp, greedy
*
* Approach: Dynamic programming with memoization or tabulation
```

```
* Time Complexity: O(n * m) where n and m are problem dimensions
* Space Complexity: O(n) or O(n * m) for DP table
*/

class Solution {
public int integerReplacement(int n) {

}
}
```

Python3 Solution:

```
"""
Problem: Integer Replacement
Difficulty: Medium
Tags: dp, greedy

Approach: Dynamic programming with memoization or tabulation
Time Complexity: O(n * m) where n and m are problem dimensions
Space Complexity: O(n) or O(n * m) for DP table
"""

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

Python Solution:

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

JavaScript Solution:

```
/**
* Problem: Integer Replacement
* Difficulty: Medium
```

```
* Tags: dp, greedy
*
* Approach: Dynamic programming with memoization or tabulation
* Time Complexity: O(n * m) where n and m are problem dimensions
* Space Complexity: O(n) or O(n * m) for DP table
*/

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

};
```

TypeScript Solution:

```
/**
* Problem: Integer Replacement
* Difficulty: Medium
* Tags: dp, greedy
*
* Approach: Dynamic programming with memoization or tabulation
* Time Complexity: O(n * m) where n and m are problem dimensions
* Space Complexity: O(n) or O(n * m) for DP table
*/

function integerReplacement(n: number): number {

};
```

C# Solution:

```
/*
* Problem: Integer Replacement
* Difficulty: Medium
* Tags: dp, greedy
*
* Approach: Dynamic programming with memoization or tabulation
* Time Complexity: O(n * m) where n and m are problem dimensions
* Space Complexity: O(n) or O(n * m) for DP table
```

```
*/

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

}
}
```

C Solution:

```
/*
* Problem: Integer Replacement
* Difficulty: Medium
* Tags: dp, greedy
*
* Approach: Dynamic programming with memoization or tabulation
* Time Complexity: O(n * m) where n and m are problem dimensions
* Space Complexity: O(n) or O(n * m) for DP table
*/

int integerReplacement(int n) {

}
```

Go Solution:

```
// Problem: Integer Replacement
// Difficulty: Medium
// Tags: dp, greedy
//
// Approach: Dynamic programming with memoization or tabulation
// Time Complexity: O(n * m) where n and m are problem dimensions
// Space Complexity: O(n) or O(n * m) for DP table

func integerReplacement(n int) int {

}
```

Kotlin Solution:

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

}
}
```

Swift Solution:

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

}
}
```

Rust Solution:

```
// Problem: Integer Replacement
// Difficulty: Medium
// Tags: dp, greedy
//
// Approach: Dynamic programming with memoization or tabulation
// Time Complexity: O(n * m) where n and m are problem dimensions
// Space Complexity: O(n) or O(n * m) for DP table

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

}
}
```

Ruby Solution:

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

end
```

PHP Solution:

```
class Solution {
```

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

}
}
```

Dart Solution:

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

}
}
```

Scala Solution:

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

}
}
```

Elixir Solution:

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

end
end
```

Erlang Solution:

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

Racket Solution:

```
(define/contract (integer-replacement n)
(-> exact-integer? exact-integer?)
)
```