Problem 1611: Minimum One Bit Operations to Make Integers Zero

Problem Information

Difficulty: Hard
Acceptance Rate: 78.45%
Paid Only: No
Tags: Dynamic Programming, Bit Manipulation, Memoization

Problem Description

Given an integer `n`, you must transform it into `0` using the following operations any number of times:

* Change the rightmost (`0th`) bit in the binary representation of `n`. * Change the `ith` bit in the binary representation of `n` if the `(i-1)th` bit is set to `1` and the `(i-2)th` through `0th` bits are set to `0`.

Return _the minimum number of operations to transform_`n` _into_`0` _._

Example 1:

Input: n = 3 **Output:** 2 **Explanation:** The binary representation of 3 is "11". "_1_ 1" -> "_0_ 1" with the 2nd operation since the 0th bit is 1. "0 _1_ " -> "0 _0_ " with the 1st operation.

Example 2:

Input: n = 6 **Output:** 4 **Explanation:** The binary representation of 6 is "110". "_1_ 10" -> "_0_ 10" with the 2nd operation since the 1st bit is 1 and 0th through 0th bits are 0. "01 _0_ " -> "01 _1_ " with the 1st operation. "0 _1_ 1" -> "0 _0_ 1" with the 2nd operation since the 0th bit is 1. "00 _1_ " -> "00 _0_ " with the 1st operation.

Constraints:

* `0 <= n <= 109`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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