5533. Minimum One Bit Operations to Make Integers Zero

issions (/contest/weekly-contest-209/problems/minimum-one-bit-operations-to-make-integers-zero/submissions/)

ontest (/contest/weekly-contest-209/)

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

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

Return *the minimum number of operations to transform* `n` *into* `0`.

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Hard

Example 1:

```
Input: n = 0
Output: 0
```

Example 2:

```
Input: n = 3
Output: 2
Explanation: The binary representation of 3 is "11".
"11" -> "01" with the 2nd operation since the 0th bit is 1.
"01" -> "00" with the 1st operation.
```

Example 3:

```
Input: n = 6
Output: 4
Explanation: The binary representation of 6 is "110".
"110" -> "010" with the 2nd operation since the 1st bit is 1 and 0th through 0th bits are
"010" -> "011" with the 1st operation.
"011" -> "001" with the 2nd operation since the 0th bit is 1.
"001" -> "000" with the 1st operation.
```

Example 4:

```
Input: n = 9
Output: 14
```

Example 5:

```
Input: n = 333
Output: 393
```

Constraints:

- $0 <= n <= 10^9$

PHP

```php
class Solution {

    /**
     * @param Integer $n
     * @return Integer
     */

    function minimumOneBitOperations($value) {
    $dec = 0;
    $bits = floor(log($value, 2));
    for ($i = $bits; $i >= 0; $i--) {
        $dec = $dec | ((($dec >> ($i + 1)) ^ ($value >> $i)) & 1) << $i);
    }
    return $dec;
}


}
```

Custom Testcase Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/404211592/)

More Details > (/submissions/detail/404211592/)

Share your acceptance!