Problem 2571: Minimum Operations to Reduce an Integer to 0

Problem Information

Difficulty: Medium
Acceptance Rate: 59.90%
Paid Only: No
Tags: Dynamic Programming, Greedy, Bit Manipulation

Problem Description

You are given a positive integer `n`, you can do the following operation **any** number of times:

* Add or subtract a **power** of `2` from `n`.

Return _the**minimum** number of operations to make _`n` _equal to_`0`.

A number `x` is power of `2` if `x == 2i` where `i >= 0` _._

Example 1:

Input: n = 39 **Output:** 3 **Explanation:** We can do the following operations: - Add 20 = 1 to n, so now n = 40. - Subtract 23 = 8 from n, so now n = 32. - Subtract 25 = 32 from n, so now n = 0. It can be shown that 3 is the minimum number of operations we need to make n equal to 0.

Example 2:

Input: n = 54 **Output:** 3 **Explanation:** We can do the following operations: - Add 21 = 2 to n, so now n = 56. - Add 23 = 8 to n, so now n = 64. - Subtract 26 = 64 from n, so now n = 0. So the minimum number of operations is 3.

Constraints:

* `1 <= n <= 105`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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