Number of 1 Bits

Difficulty	Easy
Category	Bit Manipulatin
Question	https://leetcode.com/problems/number-of-1-bits/
Solution	https://youtu.be/5Km3utixwZs
Status	Done

Question

> Write a function that takes the binary representation of an unsigned integer and returns the number of '1' bits it has (also known as the Hamming weight).
>
> **Note:**
>
> - Note that in some languages, such as Java, there is no unsigned integer type. In this case, the input will be given as a signed integer type. It should not affect your implementation, as the integer's internal binary representation is the same, whether it is signed or unsigned.
> - In Java, the compiler represents the signed integers using 2's complement notation. Therefore, in **Example 3**, the input represents the signed integer. `3`.

Example

Example 1:

```
Input: n = 00000000000000000000000000001011
Output: 3
Explanation: The input binary string00000000000000000000000000001011 has a total of three
'1' bits.
```

Example 2:

```
Input: n = 00000000000000000000000010000000
Output: 1
Explanation: The input binary string00000000000000000000000010000000 has a total of one
'1' bit.
```

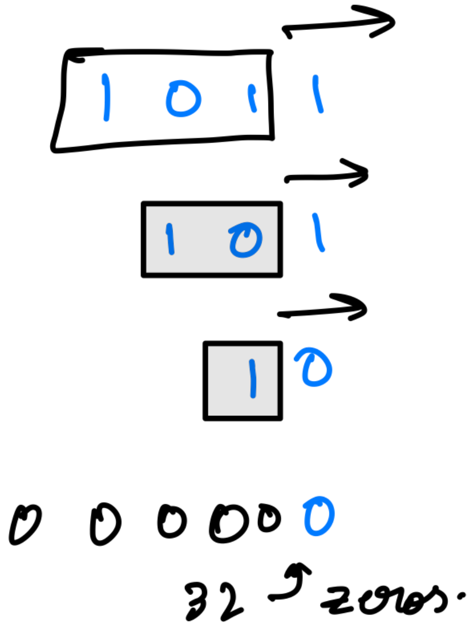
Example 3:

```
Input: n = 11111111111111111111111111111101
Output: 31
Explanation: The input binary string11111111111111111111111111111101 has a total of thirty
one '1' bits.
```

Idea

> 💡 Bitwise & with 1, if result == 1, means that bit is 1. Iterate 32 times, count++ if & == 1

Number of 1 Bits :-



1011 % 2 = 1

0 % 2 = 0

Solution

```
class Solution:
    def hammingWeight(self, n: int) -> int:
        res = 0  # Initialize a variable to store the count of set bits.
        while n:  # Continue as long as there are more bits to check.
            res += n % 2  # If the last bit of 'n' is 1, increment the result by 1.
            n = n >> 1  # Right-shift 'n' by 1 to check the next bit.

        return res  # Return the count of set bits.
----------------------------------------------------------------------
class Solution:
    def hammingWeight(self, n: int) -> int:
        res = 0
        while n:
            n &= n - 1
            res += 1
        return res
```

Explanation