# 题目

请实现一个函数，输入一个整数，输出该数二进制表示中 1 的个数。例如，把 9 表示成二进制是 1001，有 2 位是 1。因此，如果输入 9，则该函数输出 2。

**示例 1：**

**输入：**00000000000000000000000000001011

**输出：**3

**解释：**输入的二进制串 00000000000000000000000000001011 中，共有三位为 '1'。

**示例 2：**

**输入：**00000000000000000000000010000000

**输出：**1

**解释：**输入的二进制串 00000000000000000000000010000000 中，共有一位为 '1'。

**示例 3：**

**输入：**11111111111111111111111111111101

**输出：**31

**解释：**输入的二进制串 11111111111111111111111111111101 中，共有 31 位为 '1'。

# 分析

## 方法一：按位与

根据与运算定义，设二进制数字n，则有：

若n & 1 =0，则n二进制最右一位为0；

若n & 1 =1，则n二进制最右一位为1。

根据以上特点，考虑以下循环判断：

判断n最右一位是否为1，根据结果计数。

将n右移一位（本题要求把数字n看作无符号数，因此使用无符号右移操作）

注：凡是计算1的个数的都是与操作，凡是计算0的个数的都是或操作。

**代码：**

class Solution {

public:

int hammingWeight(uint32\_t n) {

int num = 0;

for(int i=0;i<32;i++){

if(n & (1<<i)) //而不是i<<1

num++;

}

return num;

}

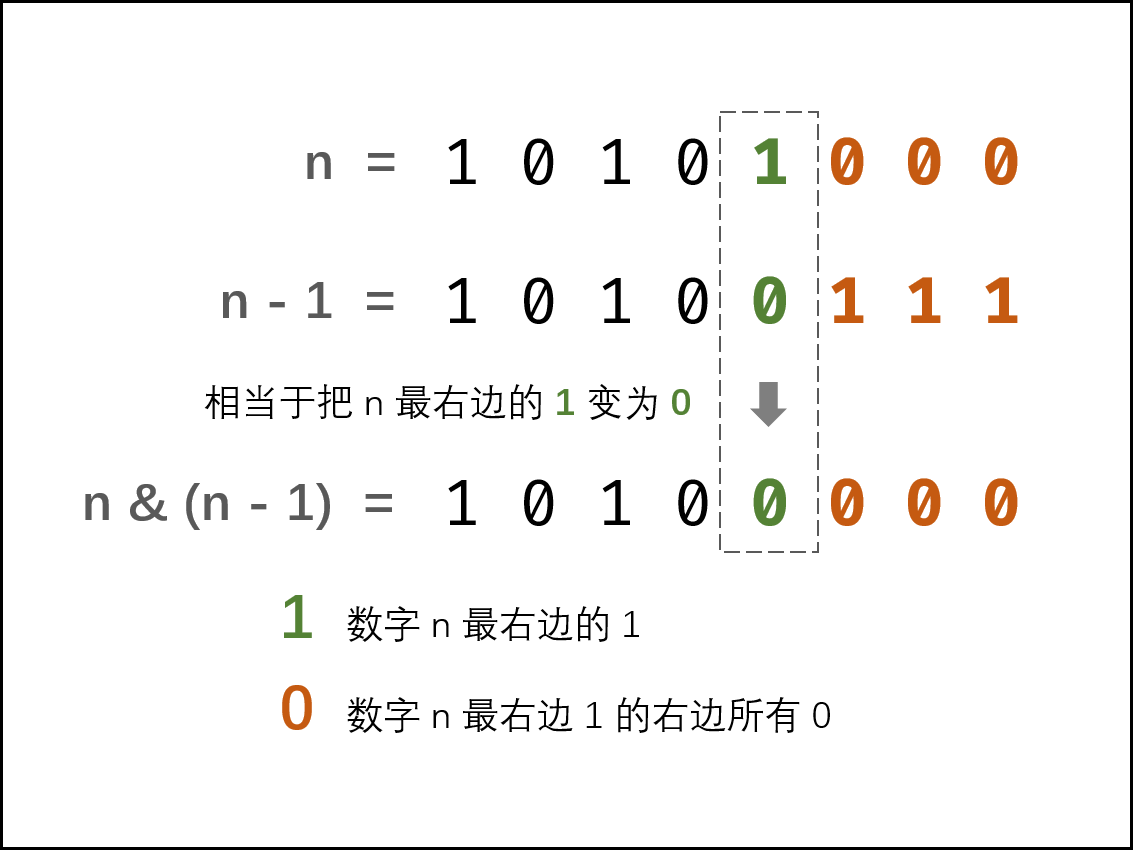
};

## 方法二：巧用n&(n−1)

思路：

(n−1) 解析：二进制数字n最右边的1变成0，此1右边的0都变成1。

n&(n−1) 解析：二进制数字n最右边的1变成0，其余不变。



1、初始化数量统计变量res。

2、循环消去最右边的1：当n=0时跳出。

2.1、res += 1：统计变量加1；

2.2、n &= n - 1：消去数字n最右边的1。

3、返回统计数量res 。

代码：

public class Solution {

public int hammingWeight(int n) {

int res = 0;

while(n != 0) {

res++;

n &= n - 1;

}

return res;

}

}