# 题目

给你一个非负整数 num ，请你返回将它变成0所需要的步数。 如果当前数字是偶数，你需要把它除以2；否则，减去1。

**示例 1：**

输入：num = 14

输出：6

解释：

步骤 1) 14 是偶数，除以 2 得到 7 。

步骤 2） 7 是奇数，减 1 得到 6 。

步骤 3） 6 是偶数，除以 2 得到 3 。

步骤 4） 3 是奇数，减 1 得到 2 。

步骤 5） 2 是偶数，除以 2 得到 1 。

步骤 6） 1 是奇数，减 1 得到 0 。

**示例 2：**

输入：num = 8

输出：4

解释：

步骤 1） 8 是偶数，除以 2 得到 4 。

步骤 2） 4 是偶数，除以 2 得到 2 。

步骤 3） 2 是偶数，除以 2 得到 1 。

步骤 4） 1 是奇数，减 1 得到 0 。

**示例 3：**

输入：num = 123

输出：12

**提示：**

0 <= num <= 10^6

# 分析

## 方法一：数学运算

class Solution {

public:

int numberOfSteps (int num) {

int ret = 0;

while(num>0)

{

if(num%2 == 0)

num /= 2;

else

num -= 1;

ret++;

}

return ret;

}

};

## 方法二：位运算

**思路：**

以32位二进制为例, 例如15(0x0000000f):

1、遇奇减1, 即将末位1变为0, 和0xfffffffe(-2)取&即可;

2、遇偶除2, 即右移一位即可;

3、两种情况都需要次数加1.

**代码：**

class Solution {

public int numberOfSteps (int num) {

int count = 0;

while (num != 0) {

count++;

num = (num & 1) == 1 ? num & -2 : num >> 1;

}

return count;

}

}