# 题目

给定一个正整数 n，你可以做如下操作：

1. 如果 n 是偶数，则用 n / 2替换 n。

2. 如果 n 是奇数，则可以用 n + 1或n - 1替换 n。

n 变为 1 所需的最小替换次数是多少？

**示例 1:**

输入:

8

输出:

3

解释:

8 -> 4 -> 2 -> 1

**示例 2:**

输入:

7

输出:

4

解释:

7 -> 8 -> 4 -> 2 -> 1

或

7 -> 6 -> 3 -> 2 -> 1

# 分析

## 方法一：位运算

**思路：**

1、偶数直接右移，只有一种选项

2、奇数+1或者-1，有两种选项。

2.1 显然，让每一步1的数目最少好处大，于是 0bxxx01 采用 -1；0bxxx11 采用+1；

2.2 特殊情况3，按上述原则+1后两次右移共需3次；减一后只需一次右移共2次，因此3采用-1操作

**代码：**

class Solution {

public:

int integerReplacement(int n) {

if(n == INT\_MAX)

return 32;

int count = 0;

while(n != 1)

{

if((n & 1) == 0)

{

n >>= 1;

}

else

{

n += ((n&2) == 0 || n == 3)?-1:1;

}

count++;

}

return count;

}

};