# 题目

给定一个整数，编写一个函数来判断它是否是 2 的幂次方。

**示例 1:**

**输入:** 1

**输出:** true

**解释:** 20 = 1

**示例 2:**

**输入:** 16

**输出:** true

**解释:** 24 = 16

**示例 3:**

**输入:** 218

**输出:** false

# 分析

## 方法一：逐次求余

class Solution {

public:

bool isPowerOfTwo(int n) {

if(0 == n)

return false;

while(0==n%2)

n = n/2;

if(1==n)

return true;

return false;

}

};

## 方法二：位运算

思路：

该问题将通过位运算在O(logN)的时间复杂度解决，通过使用如下的按位技巧：

如何获取二进制中最右边的1：x & (-x)。

如何将二进制中最右边的1设置为0：x & (x - 1)。

以下的两种解决方案背后思想都是一样的：2的幂在二进制中是有一个1后跟一些 0：

1 = (00000001) 2

2 = (00000010) 2

4 = (00000100) 2

8 = (00001000) 2

不是2的幂的二进制中有一个以上的1。

3 = (00000011) 2

5 = (00000101) 2

6 = (00000110) 2

7 = (00000111) 2

除了0，我们应该单独处理。