# 题目

给定一个整数 (32 位有符号整数)，请编写一个函数来判断它是否是 4 的幂次方。

**示例 1:**

输入: 16

输出: true

**示例 2:**

输入: 5

输出: false

**进阶：**

你能不使用循环或者递归来完成本题吗？

# 分析

## 方法一：暴力破解

## 方法二：数学运算

**思路：**

如果数字为4的幂x = 4^a，则a =log4x = 1/2log2x应为整数，那么我们检查 log2x是否为偶数就能判断x是否为4的幂。

显然，这不是我们要的答案。

**代码：**

class Solution {

public boolean isPowerOfFour(int num) {

return (num > 0) && (Math.log(num) / Math.log(2) % 2 == 0);

}

}

**复杂度：**

时间复杂度：O(1)。

空间复杂度：O(1)

## 方法三：循环

class Solution {

public:

bool isPowerOfFour(int num) {

if(num==1)

return true;

long m=1;

while(m<num)

{

m\*=4;

if(m==num)

return true;

}

return false;

}

};

## 方法四：递归

class Solution {

public:

bool isPowerOfFour(int num) {

if(num==1) return true;

else if(num==0) return false;

else return isPowerOfFour(num/4)&&num%4==0;

}

};

## 方法五：位运算

**思路：**

若一个数为4的幂，则1出现在奇数位，那么满足

1、与10101010（偶数为1，奇数为0）的32位的二进制整数相与为0

2、且1的个数为1，则说明是4的幂次

**代码：**

class Solution {

public:

bool isPowerOfFour(int num) {

bitset<32> a=num;

return !(num&2863311530)&&a.count()==1;

}

};

还有一种位运算的方法：

**思路：**

0101b用来检验奇数位1的个数，若是4的幂次，奇数位1的个数为1，反之为0

**代码：**

class Solution {

public:

bool isPowerOfFour(int num) {

if (num < 0 || num & (num-1))//排除不是2的幂次的数

{

return false;

}

return num & 0x55555555;//可以改为return num%3==1;

}

};