1、判断是不是2的幂

将2的幂写成二进制很容易看出，2的幂的二进制只有一个1，其余全是0，如下所示：

000010000...00

而将2的幂的二进制减1，其二进制变为：

000001111...11

所以判断一个数是不是2的幂的方法为使用按位与操作，如果结果为0，则是2的幂：

n & (n-1)

2、判断是不是4的幂

4的幂首先是2的幂，因为4^n = (2^2)^n，所以4的幂的二进制同样只有一个1，与2的幂不同的是，4的幂的二进制的1在偶数位上，所以判断一个数是不是4的幂的方式为：

1）首先判断是不是2的幂，使用 n & (n-1)

2）进一步判断与0x55555555的按位与结果，0x55555555是用十六进制表示的数，其奇数位上全是1，偶数位上全是0，判断 n & 0x55555555

（16进制的5对应着2进制的0101）

3、判断是不是3的幂

此方法较为通用，我们首先分析3的幂的特点，假设一个数Num是3的幂，那么所有Num的约数都是3的幂，如果一个数n小于Num且是3的幂，那么这个数n一定是Num的约数。

了解上述性质，我们只需要找到一个最大的3的幂，看看参数n是不是此最大的幂的约数就行了，假设参数是整型，那么3的最大的幂的求法为：

1. **int** maxPower = (**int**) Math.pow(3,(**int**)(Math.log(0x7fffffff)/Math.log(3)));

0x7fffffff是整型最大值，也就是Integer.maxValue()。表达式后面两个对数相处结果为double，要转化为整型。

下一步只要判断n是不是maxPower的约数即可：

maxPower % n == 0

class Solution {

public:

bool isPowerOfThree(int n) {

int maxPowerOfThree = (int)pow(3, (int)(log(0x7fffffff) / log(3)));//0x7fffffff是整型最大值，也就是Integer.maxValue()。表达式后面两个对数相处结果为double，要转化为整型。

return n>0 && maxPowerOfThree%n==0;

}

};

3的最大的幂：

假设3的n次方等于INTMAX，则n==log(3)max == lg（max）/lg(3).