# 题目

给定一个整数，写一个函数来判断它是否是 3 的幂次方。如果是，返回 true ；否则，返回 false 。

整数n是3的幂次方需满足：存在整数 x 使得 n == 3x

示例 1：

输入：n = 27

输出：true

示例 2：

输入：n = 0

输出：false

示例 3：

输入：n = 9

输出：true

示例 4：

输入：n = 45

输出：false

提示：

-231 <= n <= 231 - 1

进阶：

你能不使用循环或者递归来完成本题吗？

# 分析

## 方法一：循环迭代

找出数字n是否是数字b的幂的一个简单方法是，n%3只要余数为0，就一直将n除以b。

n=bx n=b×b×…×b

因此，应该可以将n除以b x次，每次都有0的余数，最终结果是1。

public class Solution {

public boolean isPowerOfThree(int n) {

if (n < 1) {

return false;

}

while (n % 3 == 0) {

n /= 3;

}

return n == 1;

}

}

注意最后须要一个判断，检查n == 1，否则while循环将永远不会结束。对于负数，该算法没有意义，因此我们也将包括该保护。

**复杂度分析：**

时间复杂度：O(log b(n))，在我们的例子中是O(logn)。除数是用对数表示的。

空间复杂度：O(1)，没有使用额外的空间。