

leetcode-29. 两数相除

作者: [ltw](#)

原文链接: <https://ld246.com/article/1609307066743>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

给定两个整数，被除数 dividend 和除数 divisor。将两数相除，要求不使用乘法、除法和 mod 运算。

返回被除数 dividend 除以除数 divisor 得到的商。

整数除法的结果应当截去 (truncate) 其小数部分，例如：truncate(8.345) = 8 以及 truncate(-2.735) = -2

示例 1:

输入: dividend = 10, divisor = 3

输出: 3

解释: $10/3 = \text{truncate}(3.33333..) = \text{truncate}(3) = 3$

提示:

被除数和除数均为 32 位有符号整数。

除数不为 0。

假设我们的环境只能存储 32 位有符号整数，其数值范围是 $[-2^{31}, 2^{31} - 1]$ 。本题中，如果除法果溢出，则返回 $2^{31} - 1$ 。

题解

由于负数与正数的取值范围不一致，会导致结果溢出。

此题不能使用乘法和除法，因此，可借鉴快速幂的思想。

```
class Solution {
    public int divide(int dividend , int divisor ) {
        int sign = 1;
        long x = dividend, y = divisor;
        if (x < 0 && y < 0){
            sign = 1;
            x = - x;
            y = - y;
        }
        else if (x < 0){
            sign = -1;
            x = - x;
        }
        else if (y < 0){
            sign = -1;
            y = - y;
        }
        else{
            sign = 1;
        }

        List<Long> exp = new ArrayList<>();
        for (long i = y; i <= x; i = i + i){
            exp.add(i);
        }
    }
}
```

```
long res = 0;
for (int i = exp.size() - 1; i >= 0; i--){
    if (x >= exp.get(i)){
        x -= exp.get(i);
        res += 1L << i;
    }
}
if (sign == -1){
    res = - res;
}

if (res > Integer.MAX_VALUE || res < Integer.MIN_VALUE){
    return Integer.MAX_VALUE;
}
return (int)res;
}
}
```