

0029. 两数相除

👤 [ITCharge](#) ⌚ 大约 1 分钟

- 标签：位运算、数学
- 难度：中等

题目链接

- [0029. 两数相除 - 力扣](#)

题目大意

给定两个整数，被除数 dividend 和除数 divisor。要求返回两数相除的商，并且不能使用乘法，除法和取余运算。取值范围在 $[-2^{31}, 2^{31} - 1]$ 。如果结果溢出，则返回 $2^{31} - 1$ 。

解题思路

题目要求不能使用乘法，除法和取余运算。

可以把被除数和除数当做二进制，这样进行运算的时候，就可以通过移位运算来实现二进制的乘除。

- 先将除数不断左移，移位到位数大于或等于被除数。记录其移位次数 count。
- 然后再将除数右移 count 次，模拟二进制除法运算。
 - 如果当前被除数大于等于除数，则将 1 左移 count 位，即为当前位的商，并将其累加答案上。再用除数减去被除数，进行下一次运算。

代码

py

```
class Solution:
    def divide(self, dividend: int, divisor: int) -> int:
        MIN_INT, MAX_INT = -2147483648, 2147483647
        # 标记被除数和除数是否异号
        symbol = True if (dividend ^ divisor) < 0 else False
        # 将被除数和除数转换为正数处理
        if dividend < 0:
            dividend = -dividend
        if divisor < 0:
            divisor = -divisor

        # 除数不断左移，移位到位数大于或等于被除数
        count = 0
        while dividend >= divisor:
            count += 1
            divisor <<= 1

        # 向右移位，不断模拟二进制除法运算
        res = 0
        while count > 0:
            count -= 1
            divisor >>= 1
            if dividend >= divisor:
                res += (1 << count)
                dividend -= divisor
        if symbol:
            res = -res
        if MIN_INT <= res <= MAX_INT:
            return res
        else:
            return MAX_INT
```