# 位运算实现四则运算

# 5 + 17 = ? 怎么算?

思路:

1. 每位相加不进位，此时结果为12
2. 计算进位，只有一次进位，进位值为10^1
3. 把之前两步结果相加，5 + 17 = 12 + 10 = 22

5：101　　17：10001

. 相加不进位结果：10100B

1+0 = 1，0+1=1，1+1=0，0+0=0 => 异或运算

. 进位：2^1 = 10B

0+0、0+1、1+0，不进位；1+1，进位值1<<1 => 与运算，再左移一位

. 结果相加：10100 + 10 = 10110 = 22

递归调用前两步，直到进位为0

# 推广到二进制

5：101　　17：10001

1. 相加不进位结果：10100B
   1+0 = 1，0+1=1，1+1=0，0+0=0 => 异或运算
2. 进位：2^1 = 10B
   0+0、0+1、1+0，不进位；1+1，进位值1<<1
   与运算，再左移一位
3. 结果相加：10100 + 10 = 10110 = 22
   递归调用前两步，直到进位为0

# Add

. 异或运算
. 相与，再左移一位
. 递归调用前两步，直到进位为0

```python
def add(self, num1, num2):
    # 32bits integer max
    MAX = 0x7FFFFFFF
    MASK = 0xFFFFFFFF

    ans = num1
    while num2 != 0:
        ans = (num1 ^ num2) & MASK
        num2 = ((num1 & num2) << 1) & MASK
        num1 = ans
    return ans if ans <= MAX else ~(ans ^ MASK)
```

# Subtract

正数：补码 = 原码
负数：补码 = 反码 + 1 = ~原码 + 1

so =>  a - b = a + (-b) = a + (~b + 1)

```python
def subtract(self, num1, num2):
    return self.add(num1, self.add(~num2, 1))
```

# Multiply

乘法运算:

5 * 7 = 101B * 111B = ?

```
          1 1 1      a
        1 0 1        b
-------------------
        1 1 1        b >> 0, a << 0
      0 0 0 0        b >> 1, a << 1
    1 1 1 0 0        b >> 2, a << 2
-------------------
  1 0 0 0 1 1        add
```

```python
def is_negative(self, num1, num2):
    return (num1 ^ num2) < 0

def abs(self, num):
    if num >= 0:
        return num
    else:
        return self.add(~num, 1)

def multiply(self, num1, num2):
abs1 = self.abs(num1)
abs2 = self.abs(num2)
ans = 0
while abs2 != 0:
    if abs2 & 1:
        ans = self.add(ans, abs1)
    abs2 = abs2 >> 1
    abs1 = abs1 << 1
if self.is_negative(num1, num2):
    return self.add(~ans, 1)
return ans
```

# Divide

除法运算：

37 / 5 = 100101B / 101B = ?

```
                    1 1 1
                ------------------
b 1 0 1 | 1 0 0 1 0 1        a
          1 0 1            i = 2, (a >> i) > b, a = a - b << i
        ------------------
          1 0 0 0          i = 1
            1 0 1
        ------------------
            1 1 1          i = 0
            1 0 1
        ------------------
              1 0          i < 0
```

```
def divide(self, num1, num2):
    # exception
    if num2 == 0:
        raise Exception("Divisor is zero.", num2)
    abs1 = self.abs(num1)
    abs2 = self.abs(num2)

    ans = 0
    i = 31
    while i >= 0:
        if (abs1 >> i) >= abs2:
            ans = self.add(ans, 1 << i)
            abs1 = self.subtract(abs1, abs2 << i)
        i = self.subtract(i, 1)
    if self.is_negative(num1, num2):
        return self.add(~ans, 1)
    return ans
```