# 一些多项式时间算法

# 欧几里得算法

用 $\gcd(a,b)$ 表示整数 $a$ 和 $b$ 的最大公因子.

算法 1 求最大公因子的欧几里得算法

输入　整数 $a > b \geq 0$

输出　$\gcd(a,b)$

1. if $b$ =0 return $a$

2. return (gcd($b$ , $a$ mod $b$ ))

# 扩展欧几里得算法

扩展欧几里得算法将给出 $\gcd(a,b)$ 用 $a$ 和 $b$ 表示的一个表达式.

## 算法 2 扩展欧几里得算法

输入 整数 $a > b \geq 0$

输出 整数 $\lambda, \mu$，满足 $a\lambda + b\mu = \gcd(a,b)$

1. $i \leftarrow 0;\ r_{-1} \leftarrow a;\ r_0 \leftarrow b;$

$$\lambda_{-1} \leftarrow 1;\ \mu_{-1} \leftarrow 0;\ \lambda_0 \leftarrow 0;\ \ \mu_0 \leftarrow 1;$$ （*初始化）

2. While（$r_i = a\lambda_i + b\mu_i \neq 0$）do

$$q \leftarrow r_{i-1} \div r_i;$$ （÷表示整数除法）

$$\lambda_{i+1} \leftarrow \lambda_{i-1} - q\lambda_i;\ \mu_{i+1} \leftarrow \mu_{i-1} - q\mu_i;$$

$$i \leftarrow i+1;$$

3. return $(\lambda_{i-1}, \mu_{i-1})$

# 欧几里德算法的时间复杂度

定理 1 计算最大公因子 gcd $(a,b)$至多要执行 2max$(\log a, \log b)$次模运算，因此算法 1 和算法 2 将在 2max$(\log a, \log b)$次循环内 终止.

在欧几里得算法及其扩展算法中，可以认为计算一次模运算花费一个单位的时间.

事实上，模运算的时间复杂度和除法的时间复杂度一样，它取决于两个操作数的规模. 用一个除法作为时间单位，显得太粗略. 为了更准确地度量，可以采用按位计算法来度量算术运算.

在按位计算中，所有的变量都是 0 或者 1，运算是逻辑运算而不是算术运算，即 $\wedge$, $\vee$, $\oplus$, $\neg$，分别称作与、或、异或和非.

定义 1　按位计阶号　用 $O_B(\ )$ 表示在按位计算模式下的 $O(\ )$.

在按位模式下，两个整数 $i$, $j$ 之间的加和减需要 $\max(|i|,|j|)$ 次按位运算，即时间为 $O_B(\max(|i|,|j|))$；两个整数 $i, j$ 之间的的乘和除需要 $|i|\cdot|j|$ 次按位运算，即时间为 $O_B(\log i\cdot\log j)$.

下面，我们来讨论两个欧几里得算法的时间复杂度.

算法 2.1 的递归调用次数和算法 2.2 的循环次数相同，都是 $k$ . 我们先来估计 $k$ 的大小.

首先来证明：

$$r_{j+2} < \frac{1}{2} r_j$$

由上式知，做两次带余除法可以将余数缩小一半，要得到 $\gcd(a,b)$，所要做的带余除法的次数不超过 $2\log_2 a$，即计算量为

$O_B(\log a)$.

而做一次除法的计算量为 $O_B(\log^2 a)$，因此欧几里得算法及其扩

展算法的比特计算量都为 $O_B(\log^3 a)$.

这个复杂度的估计是可以再降低的．精心实现时，注意到下面两个事实：

（1） 模运算或除法使 $a=bq+r$ 的时间代价为

$O_B(\log a \cdot \log q)$.

（2） $q_1, q_2, \cdots, q_k$ 满足 $\sum_{i=1}^{k} \log q_i = \log \prod_{i=1}^{k} q_i \le \log a$

因此，计算最大公因子的总时间不超过

$$\sum_{i=1}^{k} O_B(\log a \cdot \log q_i) \le O_B(\log^2 a)$$

以后，将用 $O_B(\log^2 a)$ 表示欧几里得算法及其扩展算法的时间复杂度.

# 模算术

定义 2　　模运算 给定整数 $x$ 和 $n>1$，定义 $x(\mathrm{mod}\ n)$ 是用 $x$ 除以 $n$ 得到的余数，即一个非负整数 $0 \le r \le n-1$，对某个整数 $k$ 满足 $x=kn+r$ .

**定理 2　模运算性质**　设整数 $x,y,n\neq 0$，模运算具有下列性质：

1．$(x+y)\bmod n=[(x\bmod n)+(y\bmod n)]\bmod n$；

2．$(-x)\bmod n=(n-x)\bmod n=n-(x\bmod n)$；（第二个等号要求 $x\bmod n\neq 0$）

3．$(x\cdot y)\bmod n=[(x\bmod n)\cdot(y\bmod n)]\bmod n$；

4．设 $\gcd(y,n)=1$，用 $y^{-1}\bmod n$ 表示 $y$ 模 $n$ 关于乘法运算的逆，它是 $[1,n-1]$ 中一个唯一确定的整数，满足

$$(y \cdot y^{-1}) \bmod n = 1.$$

与有理数中的除法一样，除以一个数的模 $n$ 定义为乘以除数的逆．与有理数中的情形一样，要求逆存在．因此，对满足 $\gcd(y, n) = 1$ 的任意 $y$，把 $x/y \bmod n$ 写成 $xy^{-1} \bmod n$.

计算 $y^{-1}$ 涉及到运用扩展欧几里得算法，所以它需要 $O_B(\log^2 n)$ 时间．因此，模 $n$ 除法的时间复杂度为 $O_B(\log^2 n)$.

由定理 2.2，模算术和整数算术非常相似．加法和乘法都服从交换律和结合律.

在模运算 $x(\bmod\ n)$ 中，商 $k$ 的值并不重要. 等式

$$x \bmod n = y \bmod n$$

表示 $x, y$ 相差 $n$ 的一个倍数，记为

$$x \equiv y \pmod{n}$$

称作 $x, y$ 模 $n$ 同余.

## 模指数

对于 $x, y < n$，模指数 $x^y \bmod n$ 按照整数幂的通常定义，$x$ 自乘 $y$ 次，给结果模 $n$．

可使用 $y-1$ 次乘法来计算 $x^y \bmod n$，但当 $y$ 很大时，这样计算的效率很低．我们下面将介绍平方-乘的方法来计算模指数．

设 $y \div 2$ 表示 $y$ 除以 2 取整，即

$$y \div 2 = \begin{cases} y/2 & \text{当}y\text{为偶数} \\ (y-1)/2 & \text{当}y\text{为奇数} \end{cases}$$

从而有：

$$x^{y}=\begin{cases}(x^{2})^{y\div 2} & \text{当}y\text{为偶数}\\ (x^{2})^{y\div 2}x & \text{当}y\text{为奇数}\end{cases}$$

上述计算给出了著名的“平方-乘”计算模指数的算法．算法重复下列步骤：将指数除以 2，执行一次平方；如果指数为奇数，要额外执行一次乘法．

## 算法 3 模指数

输入　整数 $x, y, n: x > 0, y \geq 0, n > 1;$

输出　$x^y \bmod n$

$\mathrm{mod\_exp}(x, y, n)$

1. if $y$ =0 return (1)

2. if $y \pmod 2 = 0$, return ($\mathrm{mod\_exp}(x^2 \bmod n, y \div 2, n)$);

3. return ($x \cdot \mathrm{mod\_exp}(x^2 \bmod n, y \div 2, n)$) $\bmod n$ .

考查算法 2.3 的时间复杂度. 对 $y > 0$，“除以 2”的运算恰好执行了 $\lfloor \log_2 y \rfloor + 1$ 次就得到商 0. 即递归调用 $\lfloor \log_2 y \rfloor + 1$ 次 $\mathrm{mod\_exp}(x, y, n)$，达到第 1 步中的终止条件. 每一次递归调用包括一次平方或一次平方外加一次乘法，计算时间是 $O_B(\log^2 x)$. 那么，假设 $x, y < n$，算法 2.3 的时间复杂度的上界是 $O_B(\log^3 n)$.

下表总结了基本模算术运算的时间复杂性：

| 运算 $a, b \in_u [1, n]$ | 复杂度 |
|---|---|
| $(a \pm b) \bmod n$ | $O_B(\log n)$ |
| $(a \cdot b) \bmod n$ | $O_B(\log^2 n)$ |
| $b^{-1} \bmod n$ | $O_B(\log^2 n)$ |
| $(a / b) \bmod n$ | $O_B(\log^2 n)$ |
| $a^b \bmod n$ | $O_B(\log^3 n)$ |