2022-2023学年秋季学期

课程名称: 信息安全数学基础

英文名称: Mathematical Foundations for Information Security

授课团队：胡磊、许军、王丽萍

助　　教：郭一

信息安全数学基础

Mathematical Foundations for Information Security

[第 2 次课]　整除与欧几里德算法

授课教师：胡磊、许军

授课时间：2022年8月31日、9月7日

概 要

1. 整数与多项式
2. Karatzuba快速乘法
3. 整除概念、素数与不可约多项式
4. 带余除法
5. 整数的表示
6. 幂运算方法：重复平方——乘法
7. 小步大步方法
8. 扩展欧几里德算法
9. 最大公因式与最小公倍式
10. 算术基本定理

整除概念

- 整数集合对加、减、乘运算封闭，对除法不封闭

- 这是抽象代数中环（ring）的概念

环的定义

定义: 设 R 是一个非空集合, R 上定义有两个代数运算: 加法(记为“＋”)和乘法(记为“.”), 假如

(1) $(R,+)$ 是一个**交换群**。

(2) R 关于**乘法满足结合律**。即对于任意 $a,b,c \in R$, 有

$$a \cdot (b \cdot c) = (a \cdot b) \cdot c$$

(3) **乘法对加法满足左、右分配律**, 即对于任意 $a,b,c \in R$, 有

$$a \cdot (b + c) = a \cdot b + a \cdot c,$$

$$(b + c) \cdot a = b \cdot a + c \cdot a.$$

则称 R 为环。

群的定义

定义: 设 G 是具有一个代数运算 $\circ$ 的非空集合，并且满足:

I. 结合律: $\forall a,b,c \in G$, 有 $(a \circ b) \circ c = a \circ (b \circ c)$

Ⅱ. G 中有单位元 e : $\forall a \in G, e \circ a = a \circ e = a$

Ⅲ. 对 G 中每一个元素 a, 有逆元 $a^{-1} \in G$ ，使得 $a^{-1} \circ a = a \circ a^{-1} = e$

则称 G 关于代数运算 $\circ$ 构成一个群.

具有交换律的群称为交换群或**Abelian**群

多项式

定义　　设 **R** 是域，则 **R** 上未定元 **x** 的一个多项式是形如

$$f(x)=a_nx^n+\cdots+a_2x^2+a_1x+a_0$$

的一个表达式，这里每一个 $a_i\in R$， $n\geq 0$，称 a_i 为 x^i 在 $f(x)$ 中的系数。使得 $a_m\neq 0$ 的最大整数 **m** 称为 $f(x)$ 的次数，以 $\deg f(x)$ 表示，称 a_m 为 $f(x)$ 的首项系数。如果 $f(x)=a_0$（常数多项式）且 $a_0\neq 0$，则 $f(x)$ 次数为 **0**。所有系数都为 **0** 的多项式 $f(x)$ 称为零多项式，为了方便，定义它的次数为 $-\infty$。如果 $f(x)$ 首项系数为 **1**，则称 $f(x)$ 是首一的。把 R 上的全体多项式集合记为 $R[x]$。

多项式的运算

定义　　设 $f(x)=\sum_{i=0}^{n}a_ix^i$ 和 $g(x)=\sum_{i=0}^{m}b_ix^i$ 是 **R** 上的两个多项式，定义多项式的加法和乘法如下：

（**1**） 加法。令 $M=\max\{m,n\}$，即当 $m\neq n$ 时，**M** 就是 **m** 和 **n** 中较大的那个数；当 **m=n** 时，**M=m=n**。约定

$$a_{n+1}=a_{n+2}=\cdots=a_M=0\text{，如果 } n<M\text{，}$$

$$b_{m+1}=b_{m+2}=\cdots=b_M=0\text{，如果 } m<M\text{。}$$

那么，$f(x)$ 和 $g(x)$ 可写成 $f(x)=\sum_{i=0}^{M}a_ix^i$ 和 $g(x)=\sum_{i=0}^{M}b_ix^i$，记

$$f(x)+g(x)=\sum_{i=0}^{M}(a_i+b_i)x^i\text{。}$$

多项式的运算（续）

- 多项式的乘法：

$$f(x)\cdot g(x)=a_n b_m x^{n+m}+(a_n b_{m-1}+a_{n-1}b_m)x^{n+m-1}$$
$$+\cdots+(a_1 b_0+a_0 b_1)x+a_0 b_0$$
$$=\sum_{s=0}^{m+n}\Big(\sum_{i+j=s}a_i b_j\Big)x^s$$

多项式运算律

（1） 加法交换律：

$$f(x)+g(x)=g(x)+f(x)$$

（2） 加法结合律：

$$(f(x)+g(x))+h(x)=f(x)+(g(x)+h(x))$$

（3） 乘法交换律：

$$f(x)g(x)=g(x)f(x)$$

（4） 乘法结合律：

$$(f(x)g(x))h(x)=f(x)(g(x)h(x))$$

（5） 乘法对加法的分配律：

$$f(x)(g(x)+h(x))=f(x)g(x)+f(x)h(x)$$

多项式环

- R[x]对于多项式的加法和乘法构成环，称为多项式环。

显然，$R[x]$中的多项式对于加法和乘法封闭，$x^0=1$为$R[x]$中的单位元，且对于多项式$f(x)=\sum_{i=0}^{n}a_i x^i$，有

$$-f(x)=\sum_{i=0}^{n}(-a_i)x^i \text{。}$$

- 多项式相加，对应次数不超过被加多项式的次数最大值
- 多项式乘法对应次数相加

大整数和高次多项式的快速乘法

- 加减法没有额外的快速算法
- Karatsuba乘法

Karatsuba Multiplication

- A fast algorithm for multiplication of big integers
- Most efficient way to multiply two numbers of about same magnitude
  - Assuming "+" is much cheaper than "*"
- For n-bit number
  - Ordinary "long" multiplication: $n^2 = n^{\log_2 4}$
  - Karatsuba work factor : $n^{\log_2 3} = n^{1.585}$
- Based on a simple observation…

Karatsuba Multiplication

- Consider the product

  $(a_0 + a_1 \cdot 10)(b_0 + b_1 \cdot 10)$

- Naïve approach requires 4 multiplies to determine coefficients:

  $a_0b_0 + (a_1b_0 + a_0b_1)10 + a_1b_1 \cdot 10^2$

- Same result with just 3 multiplies:

  $a_0b_0 + [(a_0 + a_1)(b_0 + b_1) - a_0b_0 - a_1b_1]10 + a_1b_1 \cdot 10^2$

Karatsuba Multiplication

- Does Karatsuba work for bigger numbers?
- For example

$$c_0 + c_1 \cdot 10 + c_2 \cdot 10^2 + c_3 \cdot 10^3 = C_0 + C_1 \cdot 10^2$$

- Where

$$C_0 = c_0 + c_1 \cdot 10 \text{ and } C_1 = c_2 + c_3 \cdot 10$$

- Can apply Karatsuba recursively to find product of numbers of any magnitude

整除概念

整数集合对加减乘运算封闭，对除法不封闭

定义1 a, b是任意两个整数，b≠0, 如果存在整数q, 使

$$a=bq$$

称b整除a或a被b整除, 记为b|a

b叫做a的因子, a叫做b的倍数

q也是a的因子, 记为a/b,

否则b不能整除a或a不能被b整除, 记 $b \nmid a.$

当 b 遍历整数 a 的所有因数时， $-b$ 遍历整数 a 的所有因数.

当 b 遍历整数 a 的所有因数时， a/b 遍历整数 a 的所有因数.

* 0 是任何非零整数的倍数． 1 是任何整数的因数．任何非零整数 a 是其自身的的倍数，也是其自身的因数．

例 2 设 a, b 为整数．若 $b|a$，则 $b\mid(-a)$, $(-b)|a$, $(-b)|(-a)$.

证　因为$b|a$，则$a=bq$

$(-a)=b(-q)$,　$a=(-b)(-q)$, $(-a)=(-b)q$

$|(-a)$,　$(-b)|(-a)$,　$(-b)\ |(-a)$

- **相差正负1。（1和-1是全部的乘法单位元）**
- **很多时候我们只考虑正整数**

定理 1 设 $a, b \neq 0, c \neq 0$ 是整数. 若 $c|b, b|a$, 则 $c|a$.(**传递性**)

证 因为$c|b$，$b|a$，有$b=cq_1, a=bq_2$

$a=bq_2=(cq_1)\, q_2=cq$

故$c|a$

定理 2 设 $a, b, c \neq 0$ 是整数. 若 $c|a, c|b$, 则 $c|a \pm b$.(**加法运算**)

证 因为$c|a$，$c|b$，有$a=cq_1, b=cq_2$

$a \pm b= cq_1 \pm cq_2 =c(q_1 \pm q_2)$

所以 $c|a \pm b$

* **定理** 3 设 $a, b, c \neq 0$ 是整数. 若 $c|a$, $c|b$, 则对任意整数 s, t, 有 $c|sa+tb$. (**整系数线性组合**)

证 因为$c|a$，$c|b$，有$a=cq_1$, $b=cq_2$

$$sa+tb= s(cq_1)+t(cq_2)=c(sq_1+tq_2)$$

所以 $c|sa+tb$

* **例** 6 设 $a, b, c \neq 0$ 是三个整数， $c|a$, $c|b$. 如果存在整数 s, t, 使得 $sa+tb=1$, 则 $c=\pm 1$.

证 $c|sa+tb=1$，所以 $c=\pm 1$

* **定理 3** 设 a, b, $c \neq 0$ 是整数. 若 $c|a$, $c|b$, 则对任意整数 s, t, 有 $c|sa+tb$. (**整系数线性组合**)

定理 4 若整数 a_1, $\ldots$, a_n 都是整数 $c \neq 0$ 的倍数, 则对任意 n 个整数 s_1, $\ldots$, s_n, 整数

$$s_1a_1 + \cdots + s_na_n$$

是 c 的倍数.

* **定理 5** 设 a, b 都是非零整数. 若 $a|b$, $b|a$, 则 $a=\pm b$.

证 因为$b|a$ ， $a|b$ ，有$a=bq_1$, $b=aq_2$

$a=aq_1q_2$, 所以 $q_1q_2=1$，$q_1=q_2=\pm 1$

故 $a=\pm b$

* **定义 2** 设整数 $n\neq 0, \pm 1$. 如果除了显然因数 ± 1 和 $\pm n$ 外, n 没有其它因数, 则 n 叫做 **素数** (或 **质数** 或 **不可约数**), 否则, n 叫做 **合数**.

因 n 和 $-n$ 同为素数或合数, 故素数总是指正整数, 通常写成 p.

多项式的整除和不可约多项式概念

- 多项式集合对加、减、乘运算封闭，对除法不封闭
- 类似地有多项式的整除和不可约多项式概念

- 多项式整除可以相差非零常数
- 非零常数是多项式环全部的乘法单位元
- 很多时候我们只考虑首一多项式

* **定理 9** (欧几里得除法) 设 a, b 是两个整数，其中 $b>0$. 则存在惟一的整数 q, r 使得

$$a = bq + r, \quad 0 \le r < b \tag{2}$$

证 存在性

$$\ldots, \ -3b, -2b, -b, 0, b, 2b, 3b, \ldots$$

$$qb \le a < (q+1)b$$

令 $r = a - qb$， 则 $a = qb + r$， $0 \le r < b$

唯一性 $a = qb + r$， $0 \le r < b$

$a = q_1 b + r_1$， $0 \le r_1 < b$, $q \ne q_1$, $r \ne r_1$

$$b \le |(q-q_1)\, b| = |-(r-r_1)| < b$$

矛盾！ 故 $q=q_1$, $r=r_1$ 。

引理 (多项式欧几里得除法) 设 $f(x) = a_n x^n + \cdots + a_1 x + a_0$ 为 n 次整系数多项式， $g(x) = x^m + \cdots + b_1 x + b_0$ 为 $m \geq 1$ 次首一整系数多项式，则存在整系数多项式 $q(x)$ 和 $r(x)$ 使得

$$f(x) = g(x)q(x) + r(x), \quad \deg r(x) < \deg g(x).$$

证 分两种情形： (I) $n < m$. 取 $q(x) = 0$, $r(x) = f(x)$, 结论成立.

(II) $n \geq m$. 对 $f(x)$ 的次数 n 作数学归纳法.

$n = m$. 我们有

$$f(x) - a_n \cdot g(x) = (a_{n-1} - a_n b_{m-1})x^{n-1} + \cdots + (a_1 - a_n b_0)x + a_0.$$

令 $q(x) = a_n$, $r(x) = f(x) - a_n \cdot g(x)$ 为所求.

比较：欧式环

- 二者是所谓的欧式环：有带余除法
  - ✓ 整数环：绝对值大小
  - ✓ 多项式环：次数
- 整数环和多项式环都是欧氏环
- 信息安全算法工作的地方

- 还有整数环和多项式环以外的欧氏环

- 整数环和多项式环，他们还有一个不同的地方：余式范围（二次互反律的高斯证明法利用了这一点）

整数带余除法的余数范围

例 2 设 m 是一个正整数. 则

i) $0, 1, \ldots, m-1$ 是模 m 的一个完全剩余系，叫做模 m 的**最小非负完全剩余系**;

ii) $1, \ldots, m-1, m$ 是模 m 的一个完全剩余系，叫做模 m 的**最小正完全剩余系**;

iii) $-(m-1),\ \ldots,\ -1,\ 0$ 是模 m 的一个完全剩余系，叫做模 m 的 **最大非正完全剩余系**；

iv) $-m,\ -(m-1),\ \ldots,\ -1$ 是模 m 的一个完全剩余系，叫做模 m 的 **最大负完全剩余系**；

v) 当 m 分别为偶数时，$-m/2,\ -(m-2)/2,\ \ldots,\ -1,\ 0,\ 1,\ \ldots,\ (m-2)/2$，或 $-(m-2)/2,\ \ldots,\ -1,\ 0,\ 1,\ \ldots,\ (m-2)/2,\ m/2$，是模 m 的一个完全剩余系；

当 m 分别为奇数时，

$$-(m-1)/2,\ \ldots,\ -1,\ 0,\ 1,\ \ldots,\ (m-1)/2$$

是模 m 的一个完全剩余系，上述两个完全剩余系统称为模 m 的一个 **绝对值最小完全剩余系**.

* **定理 6** 设 n 是一个正合数， p 是 n 的一个大于 1 的最小正因数，则 p 一定是素数，且 $p \leq \sqrt{n}$.

证 反证法：如果p不是素数，则有q，$1<q<p$, $q|p$ ，矛盾！

设 $n=pn_1$，$1<p\leq n_1<n$,

$$n=pn_1 \geq p^2$$

定理 7 设 $n>1$. 若对所有的素数 $p \leq \sqrt{n}$, 有 $p \nmid n$, 则 n 是素数.

证 用反证法

多项式也有类似的性质：

n次多项式如果不可约，则一定有次数小于等于n/2的不可约因式

定理 7 设 $n > 1$. 若对所有的素数 $p \le \sqrt{n}$, 有 $p \nmid n$, 则 n 是素数.

寻找素数的确定性方法—爱拉托斯散筛法（效率很低）：

对任意给定的正整数 N，要求出所有不超过 N 的素数．我们列出 N 个整数，从中删除 $\le \sqrt{N}$ 的所有素数 $p_1, \ldots, p_k$ 的倍数．具体地是依次删除，

$$p_1\text{的倍数}: \quad 2p_1, \ \ldots, \ [\frac{N}{p_1}]p_1;$$

$$\cdots\cdots$$

$$p_1\text{的倍数}: \quad 2p_k, \ \ldots, \ [\frac{N}{p_k}]p_k,$$

余下的整数 (不包括 1) 就是所要求的不超过 N 的素数.

例9　求出所有不超过*N*=100的素数

解　小于等于　　　　　所有素数：2，3，5，7，其倍数：

$$
\begin{array}{cccccc}
2\cdot 2, & 3\cdot 2, & 4\cdot 2, & \ldots, & 49\cdot 2, & 50\cdot 2 \\
2\cdot 3, & 3\cdot 3, & 4\cdot 3, & \ldots, & 32\cdot 3, & 33\cdot 3 \\
2\cdot 5, & 3\cdot 5, & 4\cdot 5, & \ldots, & 19\cdot 5, & 20\cdot 5 \\
2\cdot 7, & 3\cdot 7, & 4\cdot 7, & \ldots, & 13\cdot 7, & 14\cdot 7.
\end{array}
$$

对于素数 $p_1 = 2$,

1	2	3	$\cancel{4}$	5	$\cancel{6}$	7	$\cancel{8}$	9	$\cancel{10}$
11	$\cancel{12}$	13	$\cancel{14}$	15	$\cancel{16}$	17	$\cancel{18}$	19	$\cancel{20}$
21	$\cancel{22}$	23	$\cancel{24}$	25	$\cancel{26}$	27	$\cancel{28}$	29	$\cancel{30}$
31	$\cancel{32}$	33	$\cancel{34}$	35	$\cancel{36}$	37	$\cancel{38}$	39	$\cancel{40}$
41	$\cancel{42}$	43	$\cancel{44}$	45	$\cancel{46}$	47	$\cancel{48}$	49	$\cancel{50}$
51	$\cancel{52}$	53	$\cancel{54}$	55	$\cancel{56}$	57	$\cancel{58}$	59	$\cancel{60}$
61	$\cancel{62}$	63	$\cancel{64}$	65	$\cancel{66}$	67	$\cancel{68}$	69	$\cancel{70}$
71	$\cancel{72}$	73	$\cancel{74}$	75	$\cancel{76}$	77	$\cancel{78}$	79	$\cancel{80}$
81	$\cancel{82}$	83	$\cancel{84}$	85	$\cancel{86}$	87	$\cancel{88}$	89	$\cancel{90}$
91	$\cancel{92}$	93	$\cancel{94}$	95	$\cancel{96}$	97	$\cancel{98}$	99	$\cancel{100}$

对于素数 $p_2 = 3$,

1	2	3	5	7	$\cancel{9}$
11		13	$\cancel{15}$	17	19
$\cancel{21}$		23	25	$\cancel{27}$	29
31		$\cancel{33}$	35	37	$\cancel{39}$
41		43	$\cancel{45}$	47	49
$\cancel{51}$		53	55	$\cancel{57}$	59
61		$\cancel{63}$	65	67	$\cancel{69}$
71		73	$\cancel{75}$	77	79
$\cancel{81}$		83	85	$\cancel{87}$	89
91		$\cancel{93}$	95	97	$\cancel{99}$

对于素数 $p_3 = 5$,

1	2	3	5	7	
11		13		17	19
		23	~~25~~		29
31			~~35~~	37	
41		43		47	49
		53	~~55~~		59
61			~~65~~	67	
71		73		77	79
		83	~~85~~		89
91			~~95~~	97	

对于素数 $p_4 = 7$,

1	2	3	5	7	
11		13		17	19
		23			29
31				37	
41		43		47	~~49~~
		53			59
61				67	
71		73		~~77~~	79
		83			89
~~91~~				97	

不超过*N*=100

~~1~~	2	3	5	7	
11		13		17	19
		23			29
31				37	
41		43		47	
		53			59
61				67	
71		73			79
		83			89
				97	

素数的个数

有限域上低次不可约多项式寻找

- 上述原理也适用于有限域上低次不可约多项式寻找，但效率低
- 例子：二元域上4次不可约多项式寻找
  – 二元域的两个元素如何做四则运算
  – 二元域上多项式用向量表示
  – 二元域上4次不可约多项式寻找

带余除法的一个简单应用：小步大步方法

- 在讲解小步大步方法之前，讲解一下重复平方-乘方法

模重复平方计算法

$$b^n \pmod m \qquad\qquad b^n \equiv (b^{n-1} \pmod m)\cdot b \pmod m$$

$$n = n_0 + n_1 2 + \cdots + n_{k-1}2^{k-1}$$

其中 $n_i \in \{0,1\},\ i = 0,1,\ldots,k-1$. 则 $b^n \pmod m$ 的计算可归纳为

$$b^n \equiv \underbrace{\underbrace{\underbrace{b^{n_0}(b^2)^{n_1}}\cdots\cdots(b^{2^{k-2}})^{n_{k-2}}}\cdot(b^{2^{k-1}})^{n_{k-1}}} \pmod m.$$

我们最多作 $2[\log_2 n]$ 次乘法. 这个计算方法叫做 "**模重复平方计算**

具体算法如下:

$$b^n \equiv \underbrace{\underbrace{\underbrace{b^{n_0}(b^2)^{n_1}}\cdots\cdots(b^{2^{k-2}})^{n_{k-2}}}\cdot(b^{2^{k-1}})^{n_{k-1}}} \pmod{m}.$$

具体算法如下：

0). 令 $a = 1$, 并将 n 写成二进制： $n = n_0 + n_1 2 + \cdots + n_{k-1} 2^{k-1}$

1). 计算 $a_0 \equiv a \cdot b^{n_0} \pmod{m}$. 再计算 $b_1 \equiv b^2 \pmod{m}$.

2). 计算 $a_1 \equiv a_0 \cdot b_1^{n_1} \pmod{m}$. 再计算 $b_2 \equiv b_1^2 \pmod{m}$. ……

k-1). 计算 $a_{k-2} \equiv a_{k-3} \cdot b_{k-2}^{n_{k-2}} \pmod{m}$. $b_{k-1} \equiv b_{k-2}^2 \pmod{m}$.

k). 计算 $a_{k-1} \equiv a_{k-2} \cdot b_{k-1}^{n_{k-1}} \pmod{m}$.

最后， a_{k-1} 就是 $b^n \pmod{m}$.

类似算法：多项式赋值法

- 南宋数学家秦九韶
- 设x给定某个值，则有从高位到低位的计算方法：

- $a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x^1 + a_0$

  $= (\ldots((a_n x + a_{n-1})x + a_{n-2})x + \ldots + a_1)x + a_0$

- 也有从低位到高位的计算方法

Baby Step Giant Step

- n阶循环群
- Example：Know p, g and $x = g^a \pmod p$，want to find exponent a
  - n = p-1
- Let m<n
- Then $a = im + j$, some $j \in \{0,1,\ldots,m-1\}$,
  $0 < i < n/m$
- How does this help? Next slide…

Baby Step Giant Step

- Have $x = g^a \pmod p = g^{im+j} \pmod p$
- Therefore, $g^j = xg^{-im} \pmod p$
- If we find i and j so that this holds, then we have found exponent a
  - Since $a = im + j$
- How to find such i and j ?

Baby Step Giant Step

- Algorithm: Given $x = g^a \pmod p$
- **Giant steps:** Compute and store in a table, $xg^{-im} \pmod p$ for $i = 1,2,\ldots,n/m$
- **Baby steps:** Compute $g^j \pmod p$ for $j = 0,1,\ldots$ until a match with table — obtain $a = im + j$
- Expected work: at most n/m+m calculations to compute table, averagely n/m+m/2 calculations
- Storage: n/m required
- Optimal complexity: letting $m = \lceil \mathrm{sqrt}(n) \rceil$

- Spse g = 3, p = 101 and x = g^a (mod p) = 37
- Then let m = 10 and compute giant steps:

giant step i	0	1	2	3	4	5	6	7	8	9
$3^{-10i} \pmod{101}$	1	14	95	17	36	100	87	6	84	65
$37 \cdot 3^{-10i} \pmod{101}$	37	13	81	23	19	64	88	20	78	82

- **Next, compute 3^j (mod 101) until match found with last row**
- **In this case, find $3^4 = 37 \cdot 3^{-20}$ (mod 101)**
- **And we have found a = 24**

小步大步的复杂度

- ✓ 存储——时间折衷攻击
- ✓存储、时间的复杂度为群的大小的平方根
  - 二者乘积为群大小，可调准
  - 最好情形：“正方形”
- ✓ 适用任何群的平方根复杂度攻击

整数的表示

定理 1 设 $b>1$ 是正整数．则每个正整数 n 可惟一地表示成

$$n=a_{k-1}b^{k-1}+a_{k-2}b^{k-2}+\cdots+a_1b+a_0,$$

其中 a_i 是整数，$0\le a_i\le b-1$，$i=1,\ldots,k-1$，且首项系数 $a_{k-1}\neq 0$.

证　能有这样的表达式：按照带余除法连续展开（必定终止）。

展开有两种方法：

从低位到高位：依次除以b，展开商

从高位到低位：依次除以b的适当方幂，展开余数

唯一性：反证法

例 1 表示整数 642 为 2 进制.

解

$$
\begin{aligned}
642 &= 2\cdot 321 + 0, & 20 &= 2\cdot 10 + 0,\\
321 &= 2\cdot 160 + 1, & 10 &= 2\cdot 5 + 0,\\
160 &= 2\cdot 80 + 0, & 5 &= 2\cdot 2 + 1,\\
80 &= 2\cdot 40 + 0, & 2 &= 2\cdot 1 + 0,\\
40 &= 2\cdot 20 + 0, & 1 &= 2\cdot 0 + 1.
\end{aligned}
$$

因此， $642=(1010000010)_2$, 或者

$$642 = 1\cdot 2^9+0\cdot 2^8+1\cdot 2^7+0\cdot 2^6+0\cdot 2^5+0\cdot 2^4+0\cdot 2^3+0\cdot 2^2+1\cdot 2^1+0\cdot 2^0.$$

定义 1 我们用 $n=(a_{k-1}a_{k-2}\dots a_1a_0)_b$ 表示展开式:

$$n=a_{k-1}b^{k-1}+a_{k-2}b^{k-2}+\cdots+a_1b+a_0,$$

其中 $0\le a_i\le b-1,\ i=1,\dots,k-1,\ a_{k-1}\ne 0$, 并称其为整数 n 的 b **进制表示**. 这时, n 的 b- 进制位数是 $k=[\log_b n]+1$. 事实上,

$$b^{k-1}\le n<b^k \quad 或 \quad k-1\le \log_b n<k.$$

因此, $k-1=[\log_b n]$.

对于系数选取其他范围的表示也有类似结论

多项式：按某个多项式展开，类似

最大公因子

定义 1 设 $a_1, \ldots, a_n$ 是 $n\ (n \geq 2)$ 个整数．若整数 d 是它们中每一个数的因数，那么 d 就叫做 $a_1, \ldots, a_n$ 的一个 **公因数**.

如果整数 $a_1, \ldots, a_n$ 不全为零，那么整数 $a_1, \ldots, a_n$ 的所有公因数中最大的一个公因数叫做 **最大公因数**, 记作 $(a_1, \ldots, a_n)$. 特别地，当 $(a_1, \ldots, a_n) = 1$ 时，我们称 $a_1, \ldots, a_n$ **互素** 或 **互质**.

实际上， $d = (a_1, \ldots, a_n)$ 的数学表达式可叙述为:

(1) $d|a_1, \ldots, d|a_n$.　　　(2) 若 $e|a_1, \ldots, e|a_n$, 则 $e|d$.

例 1 两个整数 14 和 21 的公因数为 $\{\pm 1,\ \pm 7\}$, 它们的最大公因数 $(14, 21) = 7$.

例 4 设 $a,\ b$ 是两个整数，则 $(b,\ a) = (a,\ b)$.

例 5 设 $a,\ b$ 是两个正整数. 如果 $b|a$, 则 $(a,\ b) = b$.

例 6 设 p 是一个素数， a 为整数. 如果 $p \nmid a$, 则 p 与 a 互素.

定理 1 设 $a_1,\ \dots,\ a_n$ 是 n 个不全为零的整数，则

(i) $a_1,\ \dots,\ a_n$ 与 $|a_1|,\ \dots,\ |a_n|$ 的公因数相同;

(ii) $(a_1, \dots, a_n) = (|a_1|, \dots, |a_n|)$.

例 7 设 $a,\ b$ 是整数. 则 $(a,\ b) = (a,\ -b) = (-a,\ b) = (|a|,\ |b|)$

定理 2 设 b 是任一正整数，则 $(0, b) = b$.

* **定理 3** 设 a, b, c 是三个不全为零的整数. 如果 $a = bq + c$, 其中 q 是整数, 则 $(a, b) = (b, c)$.

证 设 $d = (a, b)$, $d' = (b, c)$, 则 $d|a$, $d|b$.

$$d|a + (-q)b = c,$$

因而, d 是 b, c 的公因数. 从而, $d \leq d'$.

同理, d' 是 a, b 的公因数, $d' \leq d$.

因此, $d = d'$.

$$(a, b) = (b, c).$$

例 10　因为 $1859=1\cdot 1537+286$

所以$(1859, 1537)=(1537, 286)$

因为 $1537=5\cdot 286+143$

所以 $(1537, 286)=(286, 143)=143$

欧几里德算法

设a, b是任意两个正整数, 记 $r_{-2}=a$, $r_{-1}=b$, 用欧几里德除法:

$$r_{-2}=r_{-1}q_0+r_0,\quad 0<r_0<r_{-1}$$

$$r_{-1}=r_0q_1+r_1,\ 0<r_1<r_0$$

$$r_0=r_1q_2+r_2,\ 0<r_2<r_1$$

$$\cdots$$

$$r_{n-3}=r_{n-2}q_{n-1}+r_{n-1},\ 0<r_{n-1}<r_{n-2}$$

$$r_{n-2}=r_{n-1}q_n+r_n,\ r_n=0$$

定理4 设a, b是任意两个正整数, 则 $(a, b)=r_{n-1}$

* **定理 3** 设 a, b, c 是三个不全为零的整数．如果 $a = bq + c$, 其中 q 是整数，则 $(a, b) = (b, c)$.

定理4　设*a*, *b*是任意两个正整数, 则 (*a*, *b*)= r_{n-1}

例 12 a=-1859, b=1573, 计算 (a, b)

解: (-1859, 1573)= (1859, 1573)

$$1859=1\cdot1537+286$$

$$1537=5\cdot 286+143$$

$$286=2\cdot 143$$

例 14　a=46480, b=39423, 计算 (a, b)

解:

$$
\begin{aligned}
46480 &= 1\cdot 39423 + 7057 \\
39423 &= 5\cdot 7057 + 4138 \\
7057 &= 1\cdot 4138 + 2919 \\
4138 &= 1\cdot 2919 + 1219 \\
2919 &= 2\cdot 1219 + 481
\end{aligned}
\qquad
\begin{aligned}
1219 &= 2\cdot 481 + 257 \\
481 &= 1\cdot 257 + 224 \\
257 &= 1\cdot 224 + 33 \\
224 &= 6\cdot 33 + 26 \\
33 &= 1\cdot 26 + 7
\end{aligned}
\qquad
\begin{aligned}
26 &= 3\cdot 7 + 5 \\
7 &= 1\cdot 5 + 2 \\
5 &= 2\cdot 2 + 1 \\
2 &= 2\cdot 1.
\end{aligned}
$$

(46480, 39423)=(39423, 4138)=(4138, 2919)

=(2919, 1219)=(1219, 481)=(481, 257)=(257, 224)

=(224, 33)=(33, 26)=(26, 7)=(7, 5)=(5, 2)=(2, 1)

=(1, 0)=1

组合系数

例　a=-1859, b=1573, 计算整数s, t 使

$$s\,a + t\,b = (a, b)$$

解: (-1859, 1573)= (1859, 1573)

$1859=1\cdot1537+286$

$1537=5\cdot 286+143$

$286=2\cdot 143$

$143 =1537-5\cdot 286$

$= 1537-5\cdot(1859- 1\cdot1537)$

$=5\cdot(-1859)+6\cdot1573$

s=5, t =6 满足

$$\boxed{s\,a + t\,b = (a, b)}$$

求组合系数:回溯方法

记 $r_{-2}=a$, $r_{-1}=b$, 用欧几里德除法:

$$r_{n-1} = r_{n-3} - r_{n-2}\, q_{n-1} = r_{n-3} - q_{n-1}\,(r_{n-4} - r_{n-3}\, q_{n-2})$$

$$= s_1 r_{n-4} - t_1 r_{n-3}$$

$$= s_2 r_{n-4} - t_2\,(r_{n-5} - r_{n-4}\, q_{n-3})$$

$$=\dots$$

$$= s_{n-1} r_{-2} - t_{n-1}\, r_{-1}$$

有s, t 满足 $s\,a + t\,b = (a, b)$

回溯方法的缺陷：来自数据结构与存储

求组合系数：正向计算

记 r_{-2} =a, r_{-1} = b, 用欧几里德除法:

$r_0 = r_{-2} - r_{-1} q_0$

$r_1 = r_{-1} - r_0 q_1 = r_{-1} - (r_{-2} - r_{-1} q_0) q_1 = s_1 r_{-2} - t_1 r_{-1}$

$r_2 = r_0 - r_1 q_2 = (r_{-2} - r_{-1} q_0) - (s_1 r_{-2} - t_1 r_{-1}) q_1 = s_2 r_{-2} - t_2 r_{-1}$

…

$r_{n-1} = r_{n-3} - r_{n-2} q_{n-1} = (s_{n-1} r_{-2} - t_{n-1} r_{-1}) - (s_{n-2} r_{-2} - t_{n-2} r_{-1}) q_{n-1}$

有 *s*, *t* 满足

$$s\,a + t\,b = (a, b)$$

The standard Euclidean algorithm proceeds by a succession of Euclidean divisions whose quotients are not used, only the remainders are kept.

For the extended algorithm, the successive quotients are used.

More precisely, the standard Euclidean algorithm with a and b as input, consists of computing a sequence $q_1, \cdots, q_k$ of quotients and a sequence $r_0, \cdots, r_{k+1}$ of remainders, such that

$$\begin{aligned} r_{-2} &= a \\ r_{-1} &= b \\ &\dots \\ r_i &= r_{i-2} - q_i r_{i-1} \quad \text{and} \quad 0 \leq r_i < |r_{i-1}| \\ &\dots \end{aligned}$$

The computation stops when one reaches a remainder r_n which is zero; the greatest common divisor is then the last non zero remainder r_{n-1}.

The extended Euclidean algorithm proceeds similarly, but adds two other sequences, as follows:

$$
\begin{array}{ccc}
a & b & r_i \\
s_{-2} = 1 & t_{-2} = 0 & r_{-2} = a \\
s_{-1} = 0 & t_{-1} = 1 & r_{-1} = b \\
s_0 = 1 & t_0 = -q_0 & r_0 \\
s_1 = -q_1 & t_1 = 1 + q_0 q_1 & r_1 \\
\vdots & \vdots & \\
s_{n-1} & t_{n-1} & r_{n-1} \neq 0 \\
s_n & t_n & r_n = 0
\end{array}
$$

$r_i = r_{i-2} - q_i r_{i-1}$ and $0 \leq r_i < r_{i-1}$ (this defines q_i)

$s_i = s_{i-2} - q_i s_{i-1}$

$t_i = t_{i-2} - q_i t_{i-1}$

$\vdots$

Proposition

(i) Let $a > b > 0$. For $0 \le i < n$, $s_i > 0 > t_i$ if i is even, and $s_i < 0 < t_i$ if i is odd.

(ii) The sequence $|s_{-1}|, |s_0|, \cdots, |s_n|$ is strictly increasing, and the sequence $|t_{-1}|, |t_0|, \cdots, |t_n|$ is also strictly increasing.

(iii) For $0 \le i \le n$,

$$\begin{pmatrix} s_{i-1} & t_{i-1} \\ s_i & t_i \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & -q_i \end{pmatrix} \begin{pmatrix} s_{i-2} & t_{i-2} \\ s_{i-1} & t_{i-1} \end{pmatrix}.$$

and for $-1 \le i \le n$, the determinant of the matrix $\begin{pmatrix} s_{i-1} & t_{i-1} \\ s_i & t_i \end{pmatrix}$ is $(-1)^{i+1}$. In particular, s_i and t_i are coprime.

(iv) $(s_n, t_n) = \pm(\frac{b}{\gcd(a,b)}, \frac{a}{\gcd(a,b)})$.

(v) For $-1 \le i < n$, $|s_i| < \frac{|b|}{\gcd(a,b)}$, $|t_i| < \frac{|a|}{\gcd(a,b)}$.

- s_{n-1}、 t_{n-1}的绝对值 分别小于$b/gcd(a,b)$和$a/gcd(a,b)$

- 通过加上或减去$(b/gcd(a,b)\ ,\ -a/gcd(a,b))$，我们可以得到 （$s_{n-1}', t_{n-1}',$），使得

$$s_{n-1}'a + t_{n-1}'b = gcd(a,b),$$

且

$$s_{n-1}' > 0 > t_{n-1}'$$

也可使得

$$s_{n-1}' < 0 < t_{n-1}'$$

- 整数欧几里德算法的迭代次数和计算复杂度：迭代次数不超过 **log a**和**log b**的最大值的两倍

模逆

- 除法：先模逆，再乘法
- 模逆：扩展欧几里德除法中只求s和t中的一个
- 扩展欧几里德除法——求(s,t)，使得sa+tb=(a,b)。
  - ✓ 求a模素数p的逆。
- 秦九韶的大衍求一术

- 完全类似地（见后面），求(s(x),t(x))，使得sf+tg=(f,g)。
  - ✓ 求f(x)模不可约多项式p(x)的逆。

注释：中国古代的大衍求一术

在《数书九章》中，秦九韶写道："大衍求一术云 置奇右上 定居右下 立天元一于左上 先以右上除右下 所得商数与左上一相生 入左下 然后乃以右行上下 以少除多 递互除之 所得商数 随即递互累乘 归左行上下 须使右上末后奇一而止 乃验左上所得 以为乘率"。书中后来又用稍微不同的语言再述之："大衍求一术云 以奇于右上 定母于右下 立天元一于左上 先以右行上下两位 以少除多 所得商数 乃递互内乘左行 使右上得一而止 左上为乘率"。

输入正整数a, m 满足$1 < a < m, \gcd(a, m) = 1$
输出正整数u 使$ua \equiv 1 \pmod{m}$.

$\begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{pmatrix} \leftarrow \begin{pmatrix} 1 & a \\ 0 & m \end{pmatrix}$;

```
while (x12 > 1) do
    if ( x22 > x12 )
        q ← ⌊(x22 − 1)/x12⌋;
        r ← x22 − q x12;
        (x11 x12; x21 x22) ← (x11 x12; q x11 + x21 r);
    if ( x12 > x22 )
        q ← ⌊(x12 − 1)/x22⌋;
        r ← x12 − q x22;
        (x11 x12; x21 x22) ← (q x21 + x11 r; x21 x22);
u ← x11;
```

while $(x_{12} > 1)$ **do**

  if $(x_{22} > x_{12})$

    $q \leftarrow \lfloor \frac{x_{22}-1}{x_{12}} \rfloor$;

    $r \leftarrow x_{22} - q x_{12}$;

    $\begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{pmatrix} \leftarrow \begin{pmatrix} x_{11} & x_{12} \\ q x_{11} + x_{21} & r \end{pmatrix}$;

  if $(x_{12} > x_{22})$

    $q \leftarrow \lfloor \frac{x_{12}-1}{x_{22}} \rfloor$;

    $r \leftarrow x_{12} - q x_{22}$;

    $\begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{pmatrix} \leftarrow \begin{pmatrix} q x_{21} + x_{11} & r \\ x_{21} & x_{22} \end{pmatrix}$;

$u \leftarrow x_{11}$;

$$\chi_0 = \begin{pmatrix} 1 & \text{奇数} \\ 0 & \text{定母} \end{pmatrix},$$

$$\chi_f = \begin{pmatrix} \text{奇数}^{-1} \pmod{\text{定母}} & 1 \\ * & * \end{pmatrix}.$$

多项式的欧几里德算法

完全类似的多项式欧几里德算法

The polynomial extended Euclidean algorithm proceeds as follows:

$$\begin{array}{lll}
a & b & r_i \\
s_{-2} = 1 & t_{-2} = 0 & r_{-2} = a \\
s_{-1} = 0 & t_{-1} = 1 & r_{-1} = b \\
s_0 = 1 & t_0 = -q_0 & r_0 \\
s_1 = -q_1 & t_1 = 1 + q_0 q_1 & r_1 \\
\vdots & \vdots & \\
s_{n-1} & t_{n-1} & r_{n-1} \neq 0 \\
s_n & t_n & r_n = 0
\end{array}$$

$$r_i = r_{i-2} - q_i r_{i-1} \quad \text{and} \quad \deg r_i(x) < \deg r_{i-1}(x)$$

$$s_i = s_{i-2} - q_i s_{i-1}$$

$$t_i = t_{i-2} - q_i t_{i-1}$$

$$\vdots$$

Proposition for polynomial Euclidean algorithm

(i) The sequence of the degrees of $s_{-1}, s_0, \cdots, s_n$ is strictly increasing, and

$$\deg s_i(x) = \deg b(x) - \deg r_{i-1}(x), \quad \forall 0 \le i \le n.$$

Similarly for the sequence $t_{-1}, t_0, \cdots, t_n$.

(ii) For $0 \le i \le n$,

$$\begin{pmatrix} s_{i-1} & t_{i-1} \\ s_i & t_i \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & -q_i \end{pmatrix} \begin{pmatrix} s_{i-2} & t_{i-2} \\ s_{i-1} & t_{i-1} \end{pmatrix}.$$

and for $-1 \le i \le n$, the determinant of the matrix $\begin{pmatrix} s_{i-1} & t_{i-1} \\ s_i & t_i \end{pmatrix}$ is $(-1)^{i+1}$. In particular, s_i and t_i are coprime.

(iii) $(s_n, t_n) = \pm(\frac{b}{\gcd(a,b)}, \frac{a}{\gcd(a,b)})$.

多项式的欧氏算法

性质1：$\deg s_{-1}(x), \deg s_0(x), \deg s_1(x), \cdots$ 严格 $\nearrow$，且

$$\deg s_i(x) = \deg b(x) - \deg r_{i-1}(x), \quad 0 \le i \le n$$

因为 $s_i = s_{i-2} - q_i s_{i-1}$，$\deg s_{i-2} < \deg s_{i-1}$，

对 $i \ge 1$，$\deg s_i = \deg(q_i s_{i-1}) = \deg q_i + \deg s_{i-1}$

$$= (\deg r_{i-2} - \deg r_{i-1}) + (\deg b - \deg r_{i-2})$$

性质2：设 $0 \le d \le \deg a - 1$，则存在唯一的 j $(-1 \le j \le n)$，使

$$\begin{cases} \deg r_j \le d, \quad \deg r_{j-1} \ge d+1 \\ \deg t_j \le \deg a - d - 1 \end{cases}$$

性质3：（译码关键方程）设

$$\begin{cases} t(x)\, b(x) = r(x) \pmod{a(x)} \\ \deg t + \deg r \le \deg a - 1 \end{cases} \quad (*)$$

则存在唯一的 j $(-1 \le j \le n)$ 和一个多项式 $\lambda(x)$，使

$$\begin{cases} t(x) = t_j(x)\lambda(x) \\ r(x) = r_j(x)\lambda(x) \end{cases}$$

因此满足 $(*)$ 式的最低次 $(t(x), r(x))$ 即为 $(t_j(x), r_j(x))$

证：设 $s(x)$ 使 $\left.\begin{array}{l} s \cdot a + t \cdot b = r \\ s_j \cdot a + t_j \cdot b = r_j \end{array}\right\} \Longrightarrow r t_j \equiv r_j t \pmod{a(x)}$

对 $d = \deg r$，由性质2

两边的次数都 $< \deg a(x)$：$\deg 左 \le \deg r + \deg a - \deg r - 1$

$\deg 右 \le \deg r + \deg t \le \deg a - 1$

$$\left.\begin{array}{l} \Longrightarrow r t_j = r_j t \Longrightarrow s t_j = s_j t \\ \gcd(s_j, t_j) = 1 \end{array}\right\} \Longrightarrow \begin{cases} s = s_j \lambda \\ t = t_j \lambda \end{cases} \text{ 对某 } \lambda(x)$$

定理6　整数a, b互素 $\Leftrightarrow$ 存在整数 s, t 满足

$$s\,a + t\,b = 1$$

证：$\Rightarrow$必要性显然。

$$s\,a + t\,b = 1，\text{则}a, b\text{互素}$$

设 d=(a, b), 则d|a, d|b, 有

$$d \mid s\,a + t\,b = 1，\text{则}d=1,\ \text{故}a, b\text{互素}$$

例 20 设四个整数 a, b, c, d 满足关系式:

$$ad - bc = 1.$$

则 $(a,b)=1$, $(a,c)=1$, $(d,b)=1$, $(d,c)=1$.

- 扩展欧几里德算法给了关于整数和多项式很多性质的存在性证明

* **定理 7** 设 a, b 是任意两个不全为零的整数，d 是正整数. 则 d 是整数 a, b 的最大公因数的充要条件是:

(i) $d|a,\ d|b$;

(ii) 若 $e|a,\ e|b$, 则 $e|d$.

证：（i）显然。

有 $s\,a + t\,b = d$

因为 $e|a$, $e|b$， 则 $e \mid s\,a + t\,b = d$

反过来，假设 (i) 和 (ii) 成立，那么

(i) 说明 d 是整数 a, b 的公因数;

(ii) 说明 d 是整数 a, b 的公因数中的最大数，因为 $e|d$ 时，有 $|e| \le d$.

因此， d 是整数 a, b 的最大公因数.

* **定理 8** 设 a, b 是任意两个不全为零的整数,

(i) 若 m 是任一正整数, 则 $(am, bm) = (a,b)m$.

(ii) 若非零整数 d 满足 $d|a$, $d|b$, 则 $(\frac{a}{d}, \frac{b}{d}) = \frac{(a,b)}{|d|}$. 特别地, $* \; (\frac{a}{(a,b)}, \frac{b}{(a,b)}) = 1$.

证: (i) 设 $d=(a, b)$, $d'=(am, bm)$

$$s\,a + t\,b = d$$

$$s\,(am) + t\,(bm) = dm$$

故 $d' \,|\, dm$ 又 $dm \,|\, d'$ 所以 $dm = d'$

(ii)

再根据 (i), 当 $d|a$, $d|b$ 时, 有

$$(a,\ b) = (\frac{a}{|d|} \cdot |d|,\ \frac{b}{|d|} \cdot |d|) = (\frac{a}{|d|},\ \frac{b}{|d|})|d| = (\frac{a}{d},\ \frac{b}{d})|d|.$$

因此, $(\frac{a}{d}, \frac{b}{d}) = \frac{(a,b)}{|d|}$. 特别地, 取 $d = (a,\ b)$, 有 $(\frac{a}{(a,b)}, \frac{b}{(a,b)}) = 1$. 故 (ii) 成立.

定理 9 设 $a_1, \ldots, a_n$ 是 n 个整数，且 $a_1 \neq 0$. 令

$$(a_1, a_2) = d_2, \quad \ldots, \quad (d_{n-1}, a_n) = d_n.$$

则 $(a_1, \ldots, a_n) = d_n$.

* **定理 9** (欧几里得除法) 设 a, b 是两个整数，其中 $b > 0$. 则存在惟一的整数 q, r 使得

$$a = bq + r, \quad 0 \leq r < b \tag{2}$$

实际运用欧几里得除法时，可根据需要将余数取成其它形式.

定理 10 (欧几里得除法) 设 a, b 是两个整数，其中 $b > 0$. 则对任意的整数 c , 存在惟一的整数 q, r 使得

$$a = bq+r, \quad c \leq r < b+c$$

引理 1 设 a, b 是两个正整数. 则 2^a-1 被 2^b-1 除的最小正余数是 2^r-1, 其中 r 是 a 模 b 的最小正余数.

证 当 $a<b$ 时, $r=a$, 结论显然成立.

当 $a\geq b$. 对 a, b 用欧几里得除法, 存在 不完全商 q 及最小正余数 r 使得

$$a=bq+r,\quad 1\leq r\leq b,$$

$$2^a-1=2^r(2^{bq}-1)+2^r-1=(2^b-1)q_1+2^r-1$$

$q_1=2^r(2^{b(q-1)}+\cdots+1)$ 为整数, 结论也成立.

引理 2 设 a, b 是两个正整数. 则 2^a-1 和 2^b-1 的最大公因数是 $2^{(a,b)}-1$.

证 运用广义欧几里得除法及引理 1 立即得到结论.

定理 10 设 a, b 是两个正整数. 则正整数 2^a-1 和 2^b-1 互素的充要条件是 a 和 b 互素.

素理想的具体例子及最小公倍数

* **定理 1** 设 a, b, c 是整数，且 $b \neq 0$, $c \neq 0$. 如果 $(a,c)=1$, 则 $(ab,c)=(b,c)$.

推论 设 a, b, c 是三个整数，且 $c \neq 0$. 如果 $c|ab$, $(a,c)=1$, 则 $c|b$.

* **定理 2** 设 p 是素数. 若 $p|ab$, 则 $p|a$ 或 $p|b$.

问题: 若 $c|ab$, 则 $c|a$ 或 $c|b$?

* **定理 3** 设 $a_1, \ldots, a_n$, c 为整数. 如果 $(a_i, c)=1$, $1 \leq i \leq n$. 则

$$(a_1 \cdots a_n, c) = 1.$$

* **推论** 设 $a_1, \ldots, a_n$ 是整数, p 是素数. 若 $p|a_1 \cdots a_n$, 则 p 一定整除某一个 a_k.

* **定义 1** 设 a_1, ..., a_n 是 n 个整数. 若 m 是这 n 个数的倍数, 则 m 叫做这 n 数的一个 **公倍数**. a_1, ..., a_n 的所有公倍数中的最小正整数叫做 **最小公倍数**, 记作 $[a_1, \ldots, a_n]$.

$m = [a_1, \ldots, a_n]$ 的数学表达式是:

(i) $a_i | m,\ 1 \leq i \leq n$; (ii) 若 $a_i | m',\ 1 \leq i \leq n$, 则 $m | m'$.

* **定理 4** 设 a, b 是两个互素正整数. 则

(i) 若 $a | m$, $b | m$, 则 $ab \mid m$; (ii) $[a, b] = ab$.

* **定理 5** 设 a, b 是两个正整数. 则

(i) 若 $a | m$, $b | m$, 则 $[a, b] \mid m$; (ii) $[a, b] = \dfrac{ab}{(a, b)}$.

证 令 $d = (a, b)$. 有 $(\frac{a}{d}, \frac{b}{d}) = 1$.

$[\frac{a}{d}, \frac{b}{d}] = \frac{a}{d} \cdot \frac{b}{d}$, 进而 $[a, b] = \frac{ab}{d}$, 即 (ii) 成立.

由 $\frac{a}{d} \mid \frac{m}{d}$, $\frac{b}{d} \mid \frac{m}{d}$, 得到 $\frac{a}{d} \cdot \frac{b}{d} \mid \frac{m}{d}$. 从而 $\frac{ab}{d} \mid m$, 即 (i) 成立.

定理 6 设 a_1, $\ldots$, a_n 是 n 个整数. 令

$$[a_1, a_2] = m_2, \ [m_2, a_3] = m_3, \ \ldots, \ [m_{n-1}, a_n] = m_n.$$

则 $[a_1, \ldots, a_n] = m_n$.

* **定理 5** 设 a, b 是两个正整数. 则

(i) 若 $a|m$, $b|m$, 则 $[a, b] \mid m$; (ii) $[a, b] = \dfrac{ab}{(a, b)}$.

* **定理 7** 设 a_1, a_2, $\ldots$, a_n 是正整数. 如果 $a_1|m$, $a_2|m$, $\ldots$, $a_n|m$, 则

$$[a_1, a_2, \ldots, a_n] \mid m$$

算术基本定理

* **定理** 1(**算术基本定理**) 任一整数 $n>1$ 都可以表示成素数的乘积, 且在不考虑乘积顺序的情况下, 该表达式是惟一的. 即

$$n = p_1 \cdots p_s, \qquad p_1 \leq \ldots \leq p_s, \tag{1}$$

p_i 是素数, 且若 $n = q_1 \cdots q_t$, $q_1 \leq \ldots \leq q_t$, q_j 是素数, 则 $s=t$, $p_i = q_i$, $1 \leq i \leq s$.

定理 2 任一整数 $n>1$ 可以惟一地表示成

$$n = p_1^{\alpha_1} \cdots p_s^{\alpha_s}, \quad \alpha_i > 0, \ i = 1, \ldots, s, \tag{3}$$

其中 $p_i < p_j$ $(i<j)$ 是素数.

(3) 式叫做 n 的 **标准分解式**.

定理 3 设 n 是大于 1 的一个整数, 且有标准分解式:

$$n = p_1^{\alpha_1} \cdots p_s^{\alpha_s}, \quad \alpha_i > 0, \ i = 1, \ldots, s,$$

则 d 是 n 的正因数当且仅当 d 有因数分解式:

$$d = p_1^{\beta_1} \cdots p_s^{\beta_s}, \quad \alpha_i \geq \beta_i \geq 0, \ i = 1, \ldots, s. \tag{4}$$

例 3 设正整数 n 有因数分解式 $n=p_1^{\alpha_1}\cdots p_s^{\alpha_s}$, $\alpha_i>0$, $i=1,\ldots,s$. 则 n 的因数个数 $d(n)=(1+\alpha_1)\cdot\cdots\cdot(1+\alpha_s)$.

证 设 $d|n, d>1$. 根据定理 3, $d=p_1^{\beta_1}\cdots p_s^{\beta_s}$, $\alpha_i\geq\beta_i\geq 0$, $i=1,\ldots,s$.

因为 β_1 的变化范围是 0 到 α_1 共 $1+\alpha_1$ 个值, $\ldots$, β_s 的变化范围是 0 到 α_s 共 $1+\alpha_s$ 个值, 所以 n 的因数个数为 $d(n)=(1+\alpha_1)\cdot\cdots\cdot(1+\alpha_s)$.

定理 4 设 a, b 是两个正整数, 且都有素因数分解式:

$$a=p_1^{\alpha_1}\cdots p_s^{\alpha_s},\qquad b=p_1^{\beta_1}\cdots p_s^{\beta_s},\quad \alpha_i\geq 0,\quad \beta_i\geq 0,\ i=1,\ldots,s,$$

则 $(a,b)=p_1^{\min(\alpha_1,\beta_1)}\cdots p_s^{\min(\alpha_s,\beta_s)}$, $[a,b]=p_1^{\max(\alpha_1,\beta_1)}\cdots p_s^{\max(\alpha_s,\beta_s)}$.

推论 设 a, b 是两个正整数, 则 $(a,b)[a,b]=ab$.

证　因为 $\min(\alpha,\beta)+\max(\alpha,\beta)=\alpha+\beta$.

例 4 计算整数 120, 150, 210, 35 的最大公因数和最小公倍数.

解

$$120 = 2^3 \cdot 3 \cdot 5, \quad 150 = 2 \cdot 3 \cdot 5^2, \quad 210 = 2 \cdot 3 \cdot 5 \cdot 7, \quad 35 = 5 \cdot 7.$$

$$(120, 150) = 2 \cdot 3 \cdot 5 = 30, \quad (30, 210) = 2 \cdot 3 \cdot 5 = 30, \quad (30, 35) = 5$$

(120, 150, 210, 35)=5

或 **$(120, 150, 210, 35)=2^0 \cdot 3^0 \cdot 5^1 \cdot 7^0=5$**

$[120, 150]=2^3 \cdot 3 \cdot 5^2=600$

$[600, 210]=2^3 \cdot 3 \cdot 5^2 \cdot 7=4200$

$[4200, 35]=2^3 \cdot 3 \cdot 5^2 \cdot 7=4200$

[120, 150, 210, 35]=4200

或 **$[120, 150, 210, 35]=2^3 \cdot 3^1 \cdot 5^2 \cdot 7^1=4200$**

例 5 设 a, b 是正整数，则存在整数 $a'|a$, $b'|b$ 使得 $a'\cdot b'=[a,b]$, $(a',b')=1$.

证 设整数 a,b 有如下的因数分解式： $a=p_1^{\alpha_1}\cdots p_s^{\alpha_s}$, $b=p_1^{\beta_1}\cdots p_s^{\beta_s}$,
其中 $\alpha_i\geq\beta_i\geq 0,\ (i=1,\ldots,t)$; $\beta_i>\alpha_i\geq 0,\ (i=t+1,\ldots,s)$.
取 $a'=p_1^{\alpha_1}\cdots p_t^{\alpha_t}$, $b'=p_{t+1}^{\beta_{t+1}}\cdots p_s^{\beta_s}$, 则整数 a', b' 即为所求.

例 6 设 $a=2^3\cdot 5^4\cdot 11^6\cdot 3^2\cdot 7^0$, $b=2^2\cdot 5^0\cdot 11^3\cdot 3^6\cdot 7^4$.
取 $a'=2^3\cdot 5^4\cdot 11^6$, $b'=3^6\cdot 7^4$, 则有

$$a'\cdot b'=2^3\cdot 5^4\cdot 11^6\cdot 3^6\cdot 7^4=[a,b].$$

- 多项式环也有算术基本定理

Q&A