离散数学

--数论和密码学

数论基础、数论算法、密码算法

2025年9月4日 星期四

总体概要

除法和模运算

质数和最大公约数

求解同余式

同余式的应用

密码学

整数除法和模算术运算

整除

定义:如果 a 和 b 是整数，且 $a \neq 0$，那么如果存在一个整数 c，使得 b=ac，则称 a 整除 b，记为$a \mid b$

① 当a整除 b时，我们称a是b的因数或除数，并且称 b是a的倍数

② 如果 $a \mid b$，那么 b/a 是一个整数

③ 如果 a 不整除 b，我们记作 $a \nmid b$

- **例题：**判断 3|7 和 3|12 是否成立。

整除性质

定理 1: a, b, c 都是整数, 且 $a \neq 0$.

i. 【线性组合性】如果 $a \mid b$ 并且 $a \mid c$, 那么 $a \mid (b + c)$;

ii. 如果 $a \mid b$，那么对于所有的整数 c, $a \mid bc$;

iii. 【传递性】如果 $a \mid b$ 并且 $b \mid c$, 那么 $a \mid c$.

iv. 【大小关系】如果$a \mid b$, $b \neq 0$,那么 $|a| \leq |b|$

证明： (i) 假设 $a \mid b$ 且 $a \mid c$ ，则存在整数 s 和 t ，使得 $b = as$ 和 $c = at$ 。所以有 $b + c = as + at = a(s + t)$ 。因此， $a \mid (b + c)$

推论：若 a, b, c 是整数，且 $a \neq 0$，且 $a \mid b$ 和 $a \mid c$，则对任意整数 m和 n，$a \mid mb+nc$。

练习: 请证明，如果$c \mid a-b$, $c \mid a'-b'$, 那么 $c \mid aa'-bb'$

除法算法

当一个整数被一个正整数除时，会得到一个商和一个余数。这通常被称为“除法算法”（Division Algorithm），但实际上它是一个定理。

除法算法:如果 a是一个整数，b 是一个正整数，那么存在唯一的整数 q 和 r，使得 $0 \leq r < b$，并且 $a = bq + r$.

- b 称为除数.
- a 称为被除数.
- q 称为商.
- r 称为余数.

Definitions of Functions **div** and **mod**

$q = a$ **div** b

$r = a$ **mod** b

例题:

- 当 101 被 11 除时，商和余数是多少？
  - 解答：当 101 被 11 除时，商为 9=101 div 11，余数为 2=101 mod 11.
- 当 −11 被 3 除时，商和余数是多少？
  - 解答：当 −11被 3 除时，商为 −4=−11 div 3−，余数为 1=−11 mod 3.

存在性的证明

良序原理(Well-Ordering Principle): 非负整数集合的任意非空子集都有一个最小元素

思考：良序原理和良序有什么联系？

Existence of q and r. Let

$$S = \{a - bk : k \in \mathbb{Z} \text{ and } a - bk \geq 0\}.$$

If $0 \in S$, then b divides a, and we can let $q = a/b$ and $r = 0$. If $0 \notin S$, we can use the Well-Ordering Principle. We must first show that S is nonempty. If $a > 0$, then $a - b \cdot 0 \in S$. If $a < 0$, then $a - b(2a) = a(1 - 2b) \in S$. In either case $S \neq \emptyset$. By the Well-Ordering Principle, S must have a smallest member, say $r = a - bq$. Therefore, $a = bq + r$, $r \geq 0$. We now show that $r < b$. Suppose that $r > b$. Then

$$a - b(q + 1) = a - bq - b = r - b > 0.$$

In this case we would have $a - b(q+1)$ in the set S. But then $a - b(q+1) < a - bq$, which would contradict the fact that $r = a - bq$ is the smallest member of S. So $r \leq b$. Since $0 \notin S$, $r \neq b$ and so $r < b$.

唯一性的证明

Uniqueness of q and r. Suppose there exist integers r, r', q, and q' such that

$$a = bq + r,\ 0 \leq r < b$$

and

$$a = bq' + r',\ 0 \leq r' < b.$$

Then $bq + r = bq' + r'$. Assume that $r' \geq r$. From the last equation we have $b(q - q') = r' - r$; therefore, b must divide $r' - r$ and $0 \leq r' - r \leq r' < b$. This is possible only if $r' - r = 0$. Hence, $r = r'$ and $q = q'$. □

Congruence Relation(同余关系)

定义:如果 a 和 b 是整数，且 m 是正整数，那么当 m 整除 $a-b$ 时，称 a 与 b 在模 m 下同余，记作 $a \equiv b \pmod{m}$.

- 符号 $a \equiv b \pmod{m}$ 表示 a 与 b 在模 m 下同余.
- 我们称 $a \equiv b \pmod{m}$ 为一个同余关系，且 m 是其模数.
- 两个整数在模 m 意义下同余当且仅当它们除以 m 后的余数相同.
- 如果 a 不与 b 在模 m 意义下同余，我们记作 $a \not\equiv b \pmod{m}$.

例题:判断 17 是否与 5 在模 6 意义下同余，24 和 14 是否在模 6 意义下同余.

解答:

- $17 \equiv 5 \pmod{6}$ 因为 6 整除 $17 - 5 = 12$.
- $24 \not\equiv 14 \pmod{6}$ 因为 $24 - 14 = 10$ 不被 6整除.

More on Congruences

定理 2:设 m 为正整数。当且仅当存在一个整数 k 使得 $a=b+km$，整数 a 和 b 在模 m 意义下同余。

证明:

- 如果 $a \equiv b \pmod{m}$，则根据同余的定义，m 整除 $a-b$。因此，存在一个整数 k，使得 $a-b=km$，即等价于 $a=b+km$.
- 反过来，如果存在一个整数 k，使得 $a=b+km$，那么 $km=a-b$。因此，m 整除 $a-b$，并且 $a \equiv b \pmod{m}$.

思考：同余关系具有什么性质？

More on Congruences

自反性: 任何正整数都和它自身同余 $a \equiv a \pmod{m}$

对称性: $a \equiv b \pmod{m} \implies b \equiv a \pmod{m}$

传递性: $a \equiv b \pmod{m}$, $b \equiv c \pmod{m} \implies a \equiv c \pmod{m}$

(mod m) 和 mod m 符号关系

在 $a \equiv b \pmod{m}$ 和 $a \bmod m = b$ 中，“mod”的用法是不同的.

- $a \equiv b \pmod{m}$ 是整数集上的一个关系，表示 a 和 b 在模 m 意义下同余.
- 在 $a \bmod m = b$ 中，"mod" 表示一个函数，该函数返回 a 除以 m 后的余数.

这两个符号之间的关系在以下定理中得到明确说明.

定理 3:设 a 和 b 为整数，m 为正整数，则 $a \equiv b \pmod{m}$ 当且仅当 $a \bmod m = b \bmod m$.

同余式的加和乘

定理 4:设 m 为正整数。如果 $a \equiv b \pmod{m}$ 且 $c \equiv d \pmod{m}$，则 $a+c \equiv b+d \pmod{m}$，$ac \equiv bd \pmod{m}$且$a^k \equiv b^k \pmod{m}$ 其中k是非负整数

证明:

- 因为 $a \equiv b \pmod{m}$ 且 $c \equiv d \pmod{m}$，根据定理 2，存在整数 s 和 t，使得 $b=a+sm$和 $d=c+tm$。
- 因此,
  - $b + d = (a + sm) + (c + tm) = (a + c) + m(s + t)$
  - $b\,d = (a + sm)(c + tm) = ac + m(at + cs + stm)$.
- 因此, $a + c \equiv b + d \pmod{m}$ and $ac \equiv bd \pmod{m}$.

例题: 因为 $7 \equiv 2 \pmod{5}$ 并且 $11 \equiv 1 \pmod{5}$,根据定理 4，可以得出

$$18 = 7 + 11 \equiv 2 + 1 = 3 \pmod{5}$$

$$77 = 7 \cdot 11 \equiv 2 \cdot 1 = 2 \pmod{5}$$

Algebraic Manipulation of Congruences (同余式的代数运算)

将有效同余的两边同时乘以一个整数会保持其有效性.

- 如果 $a \equiv b \pmod{m}$，那么 $c \cdot a \equiv c \cdot b \pmod{m}$ 也成立，其中 c是任意整数，这是通过定理 4 的 $d=c$ 得出的.

将一个整数加到有效同余的两边也会保持其有效性.

- 如果 $a \equiv b \pmod{m}$ 成立，那么 $c+a \equiv c+b \pmod{m}$ 也成立，其中 c 是任意整数，这也是通过定理 4 的 $d=c$得出的.

然而，将同余两边同时除以一个整数并不总能产生有效的同余关系.

- **例子**:同余 $14 \equiv 8 \pmod{6}$ 成立。但是将两边同时除以 2 并不会产生有效的同余，因为 14/2=7和 8/2=4，但 $7 \not\equiv 4 \pmod{6}$.

计算mod m 函数的和与积

定理4推论:设 m 为正整数，a 和 b 为整数. Then

$(a + b)$ (**mod** m) = ((a **mod** m) + (b **mod** m)) **mod** m

and

ab **mod** m = ((a **mod** m) (b **mod** m)) **mod** m.

思考：如何计算 2^{644} **mod 645**？

二进制模幂算法

在密码学中, 能够高效计算 b^n **mod** m 是非常重要的

使用n的二进制展开, $n = (a_{k-1},\ldots,a_1,a_0)_2$, 来计算b^n .

注意到: $b^n = b^{a_{k-1}\cdot 2^{k-1}+\cdots+a_1\cdot 2+a_0} = b^{a_{k-1}\cdot 2^{k-1}} \cdots b^{a_1\cdot 2} \cdot b^{a_0}.$

这个算法依次寻找b **mod** m, b^2 **mod** m, b^4 **mod** m, ..., 并且将它们相乘起来当 $a_j = 1$ 时.

procedure *modular exponentiation*(b: integer, $n = (a_{k-1}a_{k-2}\ldots a_1a_0)_2$, m: positive integers)
$x := 1$
$power := b$ **mod** m
for $i := 0$ to $k - 1$
    if $a_i = 1$ **then** $x := (x \cdot power)$ **mod** m
    $power := (power \cdot power)$ **mod** m
return x {x equals b^n **mod** m }

$O((\log m)^2 \log n)$ bit operations are used to find b^n **mod** m.

二进制模幂算法

计算 $3^{644} \bmod 645$

644的二进制展开式为 $(1010000100)_2$， k-1 = 9

首先令x=1, power= 3 **mod** 645 =3.

然后通过连续地取平方并模645来更新power值，即power: = (power • power) mod 645

a_i是645的二进制展开式的第i位.

如果a_j=1，就用x当前值乘以power 并模645来更新x值，即x : = (x • power) mod 645

具体过程如下:

二进制模幂算法

计算 $3^{644} \bmod 645$

644的二进制展开式为 $(1010000100)_2$， k-1 = 9

i=0: 因为a_0=0,所以有x=1, power=$3^2 \bmod 645=9$

i=1: 因为a_1=0,所以有x=1, power=$9^2 \bmod 645=81$

i=2: 因为a_2=1,所以有x=$1 \cdot 81 \bmod 645$, power=$81^2 \bmod 645=111$

i=3: 因为a_3=0,所以有x= 81, power=$111^2 \bmod 645=66$

i=4: 因为a_4=0,所以有x= 81, power=$66^2 \bmod 645=486$

i=5: 因为a_5=0,所以有x= 81, power=$486^2 \bmod 645=126$

二进制模幂算法

计算 $3^{644} \bmod 645$

644的二进制展开式为 $(1010000100)_2$， k-1 = 9

i=5: 因为a_5=0,所以有x= 81, power=$486^2 \bmod 645=126$

i=6: 因为a_6=0,所以有x=81, power=$126^2 \bmod 645=396$

i=7: 因为a_7=1,所以有x=$(81 \cdot 396) \bmod 645=471$,power=$396^2 \bmod 645=81$

i=8:因为a_8=0,所以有x=471,power=81^2 mod 645=111

i=9:因为a_9=1,所以有x=$(471 \cdot 111) \bmod 645=36$

根据以上步骤, 可以得出结果为$3^{644} \bmod 645=36$.

二进制模幂算法

练习：计算 $2^{644} \bmod 645$

$(644)_{10}=(1010000100)_2$

i=0: 因为a_0=0,所以有x=1, power=$2^2 \bmod 645=4$
i=1: 因为a_1=0,所以有x=1, power=$4^2 \bmod 645=16$
i=2: 因为a_2=1,所以有x=1 • 16 mod 645, power=$16^2 \bmod 645=256$
i=3: 因为a_3=0,所以有x= 16, power=$256^2 \bmod 645=391$
i=4: 因为a_4=0,所以有x= 16, power=$391^2 \bmod 645=16$
i=5: 因为a_5=0,所以有x= 16, power=$16^2 \bmod 645=256$
i=6: 因为a_6=0,所以有x=16, power=$256^2 \bmod 645=391$
i=7: 因为a_7=1,所以有x=(16 • 391) mod 645=451, power=$391^2 \bmod 645=16$
i=8: 因为a_8=0,所以有x=451, power=$16^2 \bmod 645=256$
i=9: 因为a_9=1,所以有x=(451 • 256) mod 645=1
根据以上步骤, 可以得出结果为$3^{644} \bmod 645=1$.

模m算术运算1

定义:给定 Zm 是一个包含从 0到 m–1的非负整数的集合，即 {0,1,…,m–1}

- 加法 $+_m$ 被定义为 $a +_m b = (a + b)$ **mod** m. 这是模m加法.
- 乘法$\cdot_m$ 被定义为$a \cdot_m b = (a \cdot b)$ **mod** m. 这是模m乘法.
- 进行这些运算被称为模 m 的算术运算.

例题: 计算 $7 +_{11} 9$ 和 $7 \cdot_{11} 9$.

解答: 使用上面的定义:

- $7 +_{11} 9 = (7 + 9)$ **mod** $11 = 16$ **mod** $11 = 5$
- $7 \cdot_{11} 9 = (7 \cdot 9)$ **mod** $11 = 63$ **mod** $11 = 8$

模m算术运算2

算术模运算 $+_m$ and $\cdot_m$满足许多与普通加法和乘法相同的性质

- **封闭性:** 如果a 和 b *属于* $\mathbf{Z}_m$, 那么 $a +_m b$ 和 $a \cdot_m b$ 属于 $\mathbf{Z}_m$.
- **结合律:** 如果 $a, b,$ *和* c 属于 $\mathbf{Z}_m$, 那么 $(a +_m b) +_m c = a +_m (b +_m c)$ 以及 $(a \cdot_m b) \cdot_m c = a \cdot_m (b \cdot_m c)$.
- **交换律:** 如果 a 和 b 属于$\mathbf{Z}_m$, 那么$a +_m b = b +_m a$ 以及 $a \cdot_m b = b \cdot_m a$.
- **单位元:** 0 和 1 分别是模加法和模乘法的单位元.
  - 如果 a 属于 $\mathbf{Z}_m$, 那么 $a +_m 0 = a$ 以及$a \cdot_m 1 = a$.

模m算术运算3

- **加法逆元:** 如果 $a \neq 0$ 属于 $\mathbf{Z}_m$, 那么 $m - a$ 是a的模m加法逆元，0是它自身的模m加法逆元.
  - $a +_m (m - a) = 0$ and $0 +_m 0 = 0$
- **分配律:** 如果 $a, b,$ 和 c 属于 $\mathbf{Z}_m$, 那么
  - $a \cdot_m (b +_m c) = (a \cdot_m b) +_m (a \cdot_m c)$ 以及 $(a +_m b) \cdot_m c = (a \cdot_m c) +_m (b \cdot_m c)$.

乘法逆元在模 m 的运算中不总是存在。例如，2 在模 6 的情况下没有乘法逆元，因为没有整数 x 使得 $2 \cdot x \equiv 1 \bmod 6$.

质数与最大公约数

小节概要

质数与它们的性质

关于质数的一些猜想和开放性问题

最大公约数与最小公倍数

欧几里得与扩展欧几里得算法

最大公约数的线性组合表示

质数（素数）

定义: 设 p 是大于 1 的正整数，如果 p 的正因子只有 1 和 p，那么称 p 为素数或质数 ;否则，称 p 为合数.

例:整数 7 是质数，因为它的正因数只有 1 和 7；但 9 是合数，因为它可以被 3 整除.

The Fundamental Theorem of Arithmetic (算术基本定理)

定理:每一个大于 1 的正整数都可以唯一地表示为一个素数，或者表示为两个或更多素数的乘积，其中素因数以非递减序排列.

例子:

- $641 = 641$
- $100 = 2 \cdot 2 \cdot 5 \cdot 5 = 2^2 \cdot 5^2$
- $999 = 3 \cdot 3 \cdot 3 \cdot 37 = 3^3 \cdot 37$
- $1024 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^{10}$

如何判断一个整数是不是素数？

如何找出一个整数素因子分解式？

试除法

定理: 如果a是合数， 则a必有小于等于$\sqrt{a}$的素因子

证： $a=bc$, 其中$1<b<a$, $1<c<a$. 显然, b和c中必有一个小于等于$\sqrt{a}$. 否则, $bc>(\sqrt{a})^2=a$, 矛盾.

试除法

例子 判断157和161是否是素数.

解 $\sqrt{157}$, $\sqrt{161}$都小于13, 小于13的素数有: 2, 3, 5, 7, 11.

检查结果如下:

$2 \nmid 157, 3 \nmid 157, 5 \nmid 157, 7 \nmid 157, 11 \nmid 157$

结论: 157是素数.

$2 \nmid 161, 3 \nmid 161, 5 \nmid 161, 7 \mid 161$（$161=7\times23$）

结论：161是合数.

试除法

例子 找出7007的素因子分解式

解 $7007 < 84^2$，

首先不断地用素数去除7007，从2开始。2、3和5都除不尽7007。但是，7除尽7007，7007/7 = 1001。

下一步，从7开始不断地用素数去除1001。立刻发现7还能整除1001，即1001/7 = 143。

继续从7开始不断用素数去除143。11整除143，得143/11= 13 由于13为素数，这一过程完成。

$$7007 = 7 \cdot 1001 = 7 \cdot 7 \cdot 143 = 7 \cdot 7 \cdot 11 \cdot 13 = 7^2 \cdot 11 \cdot 13$$

The Sieve of Erastosthenes (埃拉托斯特尼筛法)

试除法是一种判定整数 n 是否为素数的方法，其过程是尝试每一个小于或等于 $\sqrt{n}$ 的整数 i，并查看 n 是否能被 i 整除.

埃拉托斯特尼筛法(sieve of Eratosthenes)就是用来寻找不超过一个给定整数的所有素数

Erastothenes
(276-194 B.C.)

The Sieve of Erastosthenes (埃拉托斯特尼筛法)

例子 寻找不超过100的素数

Integers divisible by 2 other than 2 receive an underline.

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

Integers divisible by 3 other than 3 receive an underline.

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

The Sieve of Erastosthenes (埃拉托斯特尼筛法)

例子 寻找不超过100的素数

Integers divisible by 5 other than 5 receive an underline.

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

Integers divisible by 7 other than 7 receive an underline; integers in color are prime.

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

The Fundamental Theorem of Arithmetic (算术基本定理)

定理: 每一个大于 1 的正整数都可以唯一地表示为一个素数，或者表示为两个或更多素数的乘积，其中素因数以非递减序排列.

定理: 设$a>1$, 则$a=p_1^{r1}p_2^{r2}\cdots p_k^{rk}$, 其中$p_1, p_2, \cdots, p_k$是互不相同的素数， $r_1, r_2, \cdots, r_k$ 是正整数，并且在不计顺序的情况下，该表示是唯一的.

定理中的表达式称作整数 a 的素因子分解

例子： $7007 = 7^2 \cdot 11 \cdot 13$

The Fundamental Theorem of Arithmetic (算术基本定理)

推论: 设$a=p_1^{r1}p_2^{r2}\cdots p_k^{rk}$, 其中$p_1, p_2,\cdots,p_k$ 是互不相同的素数， $r_1, r_2, \cdots, r_k$ 是正整数，则正整数d为a的因子的充分必要条件是$d=p_1^{s1}p_2^{s2}\cdots p_k^{sk}$, 其中$0\le s_i\le r_i$, $i=1,2,\cdots,k$

例1：21560有多少个正因子?

解：$21560=2^3\times5\times7^2\times11$

因此 21560的正因子的个数为$4\times2\times3\times2=48$.

The Fundamental Theorem of Arithmetic (算术基本定理)

例2：10!的二进制表示中从最低位数起有多少个连续的0?

解：2, 3, 4=2^2, 5, 6=2×3, 7, 8=2^3, 9=32, 10=2×5.

10! = $2^8 \times 3^4 \times 5^2 \times 7$

故10!的二进制表示中从最低位数起有8个连续的0

练习：20!的二进制表示中从最低位数起有多少个连续的0?

Infinitude of Primes
（无穷素数）

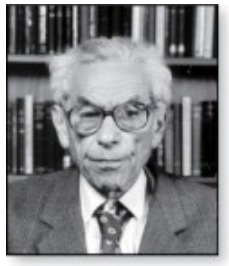
Euclid （欧几里得）
(325 B.C.E. – 265 B.C.E.)

定理: 存在无限多个素数.

证明: 假设素数的数量是有限的: $p_1, p_2, \ldots, p_n$

- Let $q = p_1 p_2 \cdots p_n + 1$
- 要么 q 是素数，要么根据算术基本定理，它是素数的乘积.
  - *但是没有任何一个* p_j 整除 q 因为如果 $p_j \mid q$, 那么 p_j 整除 $q - p_1 p_2 \cdots p_n = 1$.
  - 因此，有一个素数不在$p_1, p_2, \ldots, p_n$. 这个素数要么是 q，要么如果 q 是合数，它就是 q 的一个质因数.这与假设 $p_1,p_2,\ldots,p_n$是所有质数的集合相矛盾.
- 因此，素数的个数是无限的.

这个证明由欧几里得在《几何原本》（“The Elements”）中给出，被认为是所有数学证明中最美丽的之一。它是“天书”（The Book）中的第一个证明，这一概念灵感来自著名数学家保罗·埃尔德什（Paul Erdős）设想的完美证明集



Paul Erdős
(1913-1996)

素数的函数表示

Marin Mersenne

(1588-1648)

定义: 形如 $2^p - 1$, 其中 p 是素数，被称为梅森素数（Mersenne primes）.

- $2^2 - 1 = 3$, $2^3 - 1 = 7$, $2^5 - 1 = 37$, and $2^7 - 1 = 127$ 都是梅森素数.
- $2^{11} - 1 = 2047$ 不是梅森素数，因为$2047 = 23 \cdot 89$.
- 有一种高效的方法来确定$2^p - 1$ 是否是素数.
- 已知最大的素数是梅森素数.
- 截至 2018年早期，已发现 50 个梅森素数，其中最大的一个是 $2^{77,232,917} - 1$，这是一个有2300 万位的十进制数字
- 大互联网梅森素数搜索（Great Internet Mersenne Prime Search，GIMPS）是一个分布式计算项目，旨在寻找新的梅森素数。http://www.mersenne.org/

当p是合数时, 2^p-1一定是合数, $2^{ab}-1=(2^a-1)(2^{a(b-1)}+2^{a(b-2)}+\ldots+2^a+1)$.

素数的分布

数学家们一直对素数在正整数中的分布感兴趣。在 19 世纪，素数定理（Prime Number Theorem）被证明了，该定理给出了不超过 x 的素数个数的渐近估计.

素数定理:不超过 x 的素数个数与 $x/\ln x$的比率随着 x 的增大而趋近于 1. (其中 $\ln x$ 是 x 的自然对数)

- 该定理得知，不超过 x 的质数个数可以用 $x/\ln x$来近似.
- 随机选择一个小于 n 的正整数是素数的概率大约是 $(n/\ln n)/n = 1/\ln n$.

n	10^3	10^4	10^5	10^6	10^7
$\pi(n)$	168	1 229	9 592	78 498	664 579
$\frac{n}{\ln n}$	145	1 086	8 686	72 382	620 421
$\frac{\pi(n)}{n/\ln n}$	1.159	1.132	1.104	1.085	1.071

生成素数

生成大素数的问题在理论和实践中都具有重要意义.

我们将看到，找到有数百位数字的大素数在密码学中非常重要.

到目前为止，还没有找到一种总能生成素数的有用闭合公式。没有一个简单的函数 f(n) 能够使 f(n) 对所有正整数 n 都为素数.

然而，$f(n) = n^2 - n + 41$ 对于所有整数 1, 2, …, 40 都是素数。因为这个原因，我们可能会推测 f(n) 对所有正整数 n 都是素数。但事实上，$f(41) = 41^2$ 不是素数.

更一般地说，没有一个具有整数系数的多项式 f(n) 使得 f(n) 对所有正整数 n 都为素数.

Conjectures about Primes (关于素数的猜想)

尽管素数已经被广泛研究了几个世纪，关于它们的许多猜想仍未解决，包括:

哥德巴赫猜想：每个大于 2 的偶整数 n 都是两个素数之和。该猜想已经通过计算机验证了所有不超过 $4 \cdot 10^{18}$的正偶整数（截止2018年）。多数数学家相信这一猜想是正确的.

孪生素数猜想：孪生素数猜想认为存在无穷多对孪生素数。孪生质数是指相差 2 的素数对。例子包括 3 和 5，5 和 7，11 和 13 等。截至 2011 年中，发现的最大的孪生质数对是 $65{,}516{,}468{,}355 \cdot 2^{33{,}333} \pm 1$,它们有 100,355 位十进制数字.

Greatest Common Divisor 1 (最大公约数)

定义:设 a 和 b 为整数，且不全为零。能够同时整除 a 和 b 的最大整数 d 称为 a 和 b 的最大公约数，记作 gcd (a,b).

例: gcd(24,36)=12 , gcd(17,22)=1

最大公约数2

定义:如果两个整数 a 和 b 的最大公约数为 1，则称 a 和 b 是互素的

例子1: 17 和 22

定义: 整数$a_1, a_2, \ldots, a_n$ 是两两互素的，如果当 $1 \le i<j \le n$时有 $\gcd(a_i, a_j) = 1$.

例子2:判断整数 10、17 和 21 是否两两互素

解: 因为 gcd(10,17) = 1, gcd(10,21) = 1, and gcd(17,21) = 1, 所以10, 17, 和21 两两互素.

例子3:判断整数 10、19 和 24 是否两两互素

解: 因为 gcd(10,24) = 2, 所以10, 19, 和 24 不是两两互素

用素因子分解式找最大公约数

假设 a 和 b 的素因数分解是:

$$a = p_1^{a_1} p_2^{a_2} \cdots p_n^{a_n}, \qquad b = p_1^{b_1} p_2^{b_2} \cdots p_n^{b_n},$$

其中每个指数都是非负整数，且两个素因数分解式中出现的所有素数都包含在两者中. 那么:

$$\gcd(a,b) = p_1^{\min(a_1,b_1)} p_2^{\min(a_2,b_2)} \cdots p_n^{\min(a_n,bn)},$$

这个公式是有效的，因为等号右边的整数可以同时整除 a 和 b。没有比它更大的整数能够同时整除 a 和 b.

例子: $120 = 2^3 \cdot 3 \cdot 5$ $500 = 2^2 \cdot 5^3$

解: $\gcd(120,500) = 2^{\min(3,2)} \cdot 3^{\min(1,0)} \cdot 5^{\min(1,3)} = 2^2 \cdot 3^0 \cdot 5^1 = 20$

使用素因数分解式来求两个正整数的最大公约数并不高效，因为目前没有高效的算法来找出正整数的素因数分解.

Least Common Multiple (最小公倍数)

定义:正整数 a 和 b 的最小公倍数是同时能被 a 和 b 整除的最小正整数，记作 lcm(a,b).

最小公倍数也可以通过素因数分解来计算.

$$\text{lcm}(a,b) = p_1^{\max(a_1,b_1)} p_2^{\max(a_2,b_2)} \dots p_n^{\max(a_n,bn)},$$

这个数字可以被 a和 b 同时整除，并且没有更小的数字可以同时被 a 和 b 整除.

例子: lcm($2^3 3^5 7^2$, $2^4 3^3$) = $2^{\max(3,4)}\ 3^{\max(5,3)}\ 7^{\max(2,0)} = 2^4\ 3^5\ 7^2$

两个整数的最大公约数和最小公倍数之间的关系是:

定理: a，b为正整数. 那么ab = gcd(a,b) ·lcm(a,b)

Euclidean Algorithm[1] (欧几里得算法)

欧几里得算法是一种高效计算两个整数最大公约数的方法。其基于的理念是，当 $a>b$ 且 c 是 a 除以 b 的余数时，gcd(a,b) 等于 gcd(b,c).

Euclid
(325 B.C.E. – 265 B.C.E.)

例子: 找到 gcd(91, 287):

- $287 = 91 \cdot 3 + 14$ Divide 287 by 91
- $91 = 14 \cdot 6 + 7$ Divide 91 by 14
- $14 = 7 \cdot 2 + 0$ Divide 14 by 7

Stopping condition

gcd(287, 91) = gcd(91, 14) = gcd(14, 7) = 7

Euclidean Algorithm[1] (欧几里得算法)

欧几里得算法用伪代码表示如下:

```
procedure gcd(a, b: positive integers)
x := a   y := b
while   y ≠ 0
      r := x mod y
      x := y
      y := r
return x {gcd(a,b) is x}
```

欧几里得算法的正确性[1]

引理: 令 $a = bq + r$, 其中 a, b, q, 和 r 都是整数. 那么 $\gcd(a,b) = \gcd(b,r)$.

证明:

- 假设 d 同时整除 a 和 b。那么 d 也整除 $a-bq=r$。因此，a 和 b 的任何公约数也必须是 b 和 r 的公约数.
- 假设 d 同时整除 b 和 r。那么 d 也整除 $bq+r=a$。因此，b 和 r 的任何公约数也必须是 a 和 b 的公约数.
- 因此, $\gcd(a,b) = \gcd(b,r)$.

欧几里得算法的正确性2

假设 a 和 b 是正整数且 $a \geq b$。设 $r_0=a$和 $r_1=b$。通过连续应用除法算法，我们得到如下结果:

$$
\begin{aligned}
r_0 &= r_1 q_1 + r_2 && 0 \leq r_2 < r_1, \\
r_1 &= r_2 q_2 + r_3 && 0 \leq r_3 < r_2, \\
&\vdots \\
r_{n-2} &= r_{n-1} q_{n-1} + r_n && 0 \leq r_n < r_{n-1}, \\
r_{n-1} &= r_n q_n.
\end{aligned}
$$

$$
\begin{aligned}
2415 &= 945 \cdot 2 + 525 \\
945 &= 525 \cdot 1 + 420 \\
525 &= 420 \cdot 1 + 105 \\
420 &= 105 \cdot 4 + 0.
\end{aligned}
$$

最终，余数为零会出现在序列中: $a = r_0 > r_1 > r_2 > \cdots \geq 0$.这个序列中最多不能包含超过 a 项

根据引理1

$\gcd(a,b) = \gcd(r_0,r_1) = \cdots = \gcd(r_{n-1},r_n) = \gcd(r_n, 0) = r_n$.

因此，最大公约数是这个除法序列中的最后一个非零余数.

最大公约数表示成一个线性组合

定理:如果 a 和 b 是任意整数，且不全为零，那么 gcd(a,b)是集合{$ax + by : x, y \in \mathbb{Z}$} 中的最小正整数元素.

证明:

设 s 是 a 和 b 的最小正线性组合.

设 q 是 a 除以 s 的商. 那么 **a mod s**=a-qs=a-q(ax+by)=a(1-qx)+b(-qy).

因此，a mod s是 a 和 b 的线性组合.

由于 s 是所有线性组合中的最小正数，并且0<=**a mod s** <s，

a mod s不能是正数. 因此 **a mod s**=0.

这意味着 s 是 a 的因子，同样 s 也是 b 的因子。因此 s 是 a 和 b 的公约数, gcd(a,b)>=s.

由于 gcd(a,b) 同时整除 a 和 b，且 s 是 a 和 b 的线性组合, 我们有 gcd(a,b) | s. 但是 gcd(a,b) | s 并且 s>0 意味着 gcd(a,b)<=s.

推论: 对于整数 a 和 b, 如果 d |a 并且 d|b, 那么 d|gcd(a,b).

最大公约数表示成一个线性组合

Étienne Bézout

(1730-1783)

Bézout's Theorem (贝祖定理):如果 a 和 b 是正整数，那么存在整数 s 和 t 使得 gcd $(a,b)=sa+tb$.

定义:如果 a 和 b 是正整数，那么使得 gcd $(a,b)=sa+tb$的整数 s 和 t 称为 a 和 b 的 贝祖系数.

根据 贝祖定理，整数 a 和 b 的最大公约数可以表示为 $sa+tb$，其中 s 和 t 是整数。这是 a 和 b 的一个线性组合，其系数为整数.

- gcd(6,14) = (−2)·6 + 1·14

求贝祖系数

例子: 将 gcd(252,198)=18 表示为 252 和 198 的线性组合.

解: 首先使用欧几里得算法证明 gcd(252,198)=18

i. $252 = 1 \cdot 198 + 54$
ii. $198 = 3 \cdot 54 + 36$
iii. $54 = 1 \cdot 36 + 18$
iv. $36 = 2 \cdot 18$

- 现在从上面第（iii）式和第（i）式反向推导
  - $18 = 54 - 1 \cdot 36$
  - $36 = 198 - 3 \cdot 54$
- 将第二个方程代入第一个方程中:
  - $18 = 54 - 1 \cdot (198 - 3 \cdot 54) = 4 \cdot 54 - 1 \cdot 198$
- 接着将 54=252−1·198 (由第(i)式得) 代入上式:
  - $18 = 4 \cdot (252 - 1 \cdot 198) - 1 \cdot 198 = 4 \cdot 252 - 5 \cdot 198$

这个方法展示了“两步法”，先使用欧几里得算法找到最大公约数，然后通过回代的方式将最大公约数表示为原始两个整数的线性组合。还有一种称为扩展欧几里得算法的“单步法”，可以在进行欧几里得算法的同时贝祖系数，将最大公约数直接表示为线性组合(扩展欧几里得算法).

扩展欧几里得算法

$\gcd(a,b) = r_n$

设 $r_0=a$和 $r_1=b$

$$r_0 = r_1 q_1 + r_2$$

$$r_1 = r_2 q_2 + r_3$$

$$\vdots$$

$$r_{n-2} = r_{n-1} q_{n-1} + r_n$$

$$r_{n-1} = r_n q_n.$$

$s_0=1,\ s_1=0 \qquad t_0=0,\ t_1=1$

$s_i=s_{i-2}-s_{i-1}q_{i-1} \qquad t_i=t_{i-2}-t_{i-1}q_{i-1} \qquad (i= 1, 2, \dots, n)$

$as_i+bt_i = r_i \qquad (i= 1, 2, \dots, n)$

$as_n+bt_n = r_n$

已知 $\gcd(a,b) = r_n$

所以 $as_n+bt_n = \gcd(a,b)$

所以 $s= s_n, \qquad t=t_n$

扩展欧几里得算法

例子: 将 gcd(100,35) 表示为 100 和 35 的线性组合.

解: r_0=100 和 r_1=35

100 = 35 × 2 + 30

35 = 30 × 1 + 5

30 = 5 × 6

s_0=1, s_1=0 t_0=0, t_1=1

$s_2= s_0-s_1q_1= 1-0 \times 2 = 1$ $t_2= t_0-t_1q_1= 0-1 \times 2 = -2$

$s_3= s_1-s_2q_2= 0-1 \times 1 = -1$ $t_3= t_1-t_2q_2= 1-(-2) \times 1 = 3$

$gcd(100,35)= 100s_3+35t_3= 100 \times (-1) + 35 \times 3$

Dividing Congruences by an Integer

将有效同余式的两边同时除以一个整数，并不总是能得到一个有效的同余式。

但如果这个整数与模数互素，则可以得到一个有效的同余式:

定理 :设 m 为正整数，a、b 和 c 为整数. 如果 $ac \equiv bc \pmod{m}$ 并且 $\gcd(c,m) = 1$, 那么 $a \equiv b \pmod{m}$.

证明: 因为 $ac \equiv bc \pmod{m}$, 那么$m \mid ac - bc = c(a - b)$ 由于 $\gcd(c,m) = 1$, 所以 $m \mid a - b$. 因此, $a \equiv b \pmod{m}$.