西安电子科技大学 XIDIAN UNIVERSITY
计算机科学与技术学院 School of Computer Science and Technology
国家示范性软件学院 National Pilot School of Software Engineering

计算机安全导论

第 4 章+
数论基础知识

主讲人：张志为

二〇二四年秋季学期

提纲

提纲

整数

整数

整数集{… , -3, -2, -1, 0, 1, 2, 3, …} 记为Z。

整除

设a, b为整数。若存在某个整数c, 使得b=ac, 则称a整除b（等价地，称a是b的一个因子，或者说a为b的一个因子）。若a整除b，则记为 a | b 。

例如

整数

整除的基本性质

对所有的整数a，b，c有以下正确结论：

- a | a
- 若 a | b 且 b | c ，则 a | c
- 若 a | b 且 a | c，则对于所有的整数x，y，有a |（bx+cy）
- 若 a | b 且 b | a，则 a = + b 或 a = -b

例如

整数

□ 整数的整除算法

若a，b均为整数，且b>=1, 则按照a除以b的普通长除法可以找到整数q（商）和r余数，使得：a=qb+r，

其中0<=r<b。

且q和r唯一。

除法所得余数记为a mod b，商记为a div b。

□ 例如

整数

□ 整数模n

设n为一整数。

若a，b为整数，则称a与b是模n同余的，记为$a \equiv b \pmod{n}$。

□ 同余的性质：对所有的整数a，a1，b，b1，c有：

$a \equiv b \pmod{n}$当且仅当a与b被n除时所得的余数相同

自反性：$a \equiv a \pmod{n}$

对称性：若$a \equiv b \pmod{n}$，则$b \equiv a \pmod{n}$

传递性：若$a \equiv b \pmod{n}$，$b \equiv c \pmod{n}$，则$a \equiv c \pmod{n}$

若$a \equiv a1 \pmod{n}$，$b \equiv b1 \pmod{n}$，则$a + b \equiv a1 + b1 \pmod{n}$，且$a\ b \equiv a1\ b1 \pmod{n}$

整数

□ 整数的乘法逆元

设a为整数，若存在整数x，使得$ax \equiv 1 \pmod{n}$，则称x为a的模n的乘法逆元。

若x存在，则它是唯一的，此时称a为可逆的，a的逆元记为a^{-1}。

设a为整数，则a可逆当且仅当 gcd（a，n）= 1。

乘法逆元

□ 若ab≡1 mod n， 则a和b互为mod n的乘法逆元。

例：2*4 ≡1 mod 7，

则2是4模7的乘法逆元

或 4是2模7的乘法逆元。

求乘法的逆元用**扩展欧几里得算法**（Extended Euclidean algorithm），扩展欧几里得算法可用于RSA加密等领域。

乘法逆元

欧几里得算法（Euclidean algorithm）

欧几里得（古希腊数学家）算法，又称为辗转相除法，是用来求两个正整数最大公约数（GCD, Greatest Common Divisor）的算法。

假如：用欧几里得算法求 1997 和 615 两个正整数的最大公约数,:

1997 / **615** = 3 (余 152)

615 / 152 = 4 (余7)

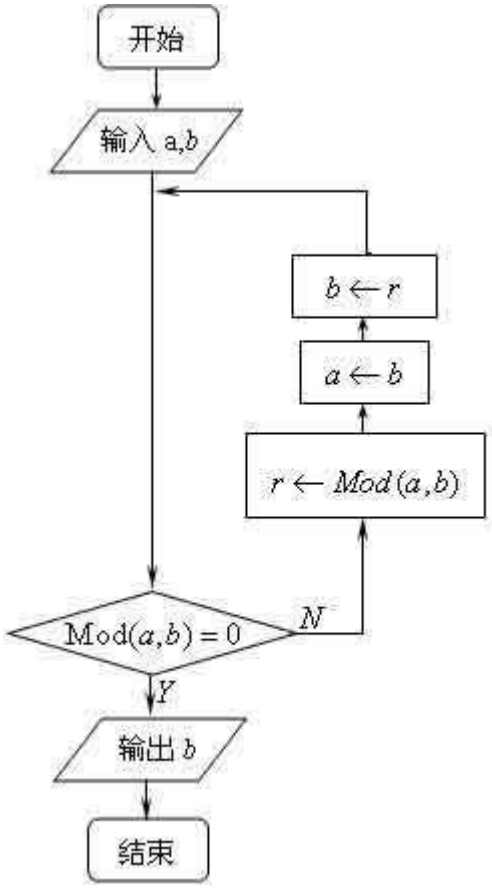
152 / 7 = 21(余5)

7 / 5 = 1 (余2)

5 / 2 = 2 (余1)

2 / 1 = 2 (余0)

至此，**最大公约数为1**




```
int gcd(int a,int b)
{
    if(b==0) return a;
    else return gcd(b,a%b);
}
```

以除数和余数反复做除法运算，当余数为 0 时，取当前算式除数为最大公约数，所以就得出了 1997 和 615 的最大公约数 1。

乘法逆元

- **定理**：对于不完全为 0 的非负整数 a，b，gcd（a，b）表示 a，b 的最大公约数，必然存在整数

  对 x，y，使得 gcd（a，b）=a*x+b*y。
  - 当 b=0 时，gcd(a,b)=a，此时 x=1 , y=0
  - 当 b!=0 时，
  - 设 $ax_1+by_1=gcd(a,b)=gcd(b,a\%b)=bx_2+(a\%b)y_2$
  - 又因 a%b=a-a/b*b
  - 则 $ax_1+by_1=bx_2+(a-a/b*b)y_2$
  - $ax_1+by_1=bx_2+ay_2-a/b*by_2$
  - $ax_1+by_1=ay_2+bx_2-b*a/b*y_2$
  - $ax_1+by_1=ay_2+b(x2-a/b*y_2)$
  - 解得 $x_1=y_2$, $y_1=x_2-a/b*y_2$
  - 因为当 b=0 时存在 x , y 为最后一组解
  - 而每一组的解可根据后一组得到
  - 所以第一组的解 x , y 必然存在
  - 得证

```
int e_gcd(int a,int b,int &x,int &y)
{
    if(b==0)
    {
        x=1;
        y=0;
        return a;
    }
    int ans=e_gcd(b,a%b,x,y);
    int temp=x;
    x=y;
    y=temp-a/b*y;
    return ans;
}
```

- 扩展欧几里德算法的主要有以下三方面应用
  - 求解不定方程
  - 求解模线性方程（线性同余方程）
  - 求解模的逆元

乘法逆元

□ 扩展欧几里得算法EUCLID(*m*, *b*)

1.(A1, A2, A3)←(1, 0, m);

(B1, B2, B3)←(0, 1, b)

2. if B3 = 0

return A3 = gcd(m, b); no inverse

3. if B3 = 1

return B3 = gcd(m, b); B2 = b–1 mod m

4. Q = $\lfloor A3/B3 \rfloor$

5. (T1, T2, T3) = (A1 – Q B1, A2 – Q B2, A3 – Q B3)

6. (A1, A2, A3) = (B1, B2, B3)

7. (B1, B2, B3) = (T1, T2, T3)

8. goto 2

乘法逆元

求550模1759的乘法逆元，即gcd(1759, 550)=1

Q	A1	A2	A3	B1	B2	B3
—	1	0	1759	0	1	550
3	0	1	550	1	–3	109
5	1	–3	109	–5	16	5
21	–5	16	5	106	–339	4
1	106	–339	4	–111	355	1

T1 = A1 – Q B1

T2 = A2 – Q B2

T3 = A3 – Q B3

(A1, A2, A3) = (B1, B2, B3)

(B1, B2, B3) = (T1, T2, T3)

提纲

素数

- **数论主要关心的是素数**
- **整数p > 1是素数，当且仅当它只有因子±1和± p**

素数不能写作其它数的乘积

1是素数，但一般对它没兴趣

- **例如：2, 3, 5, 7是素数，4, 6, 8, 9, 10 不是素数**
- **200以内的素数**

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 101 103 107 109 113 127 131 137 139 149 151

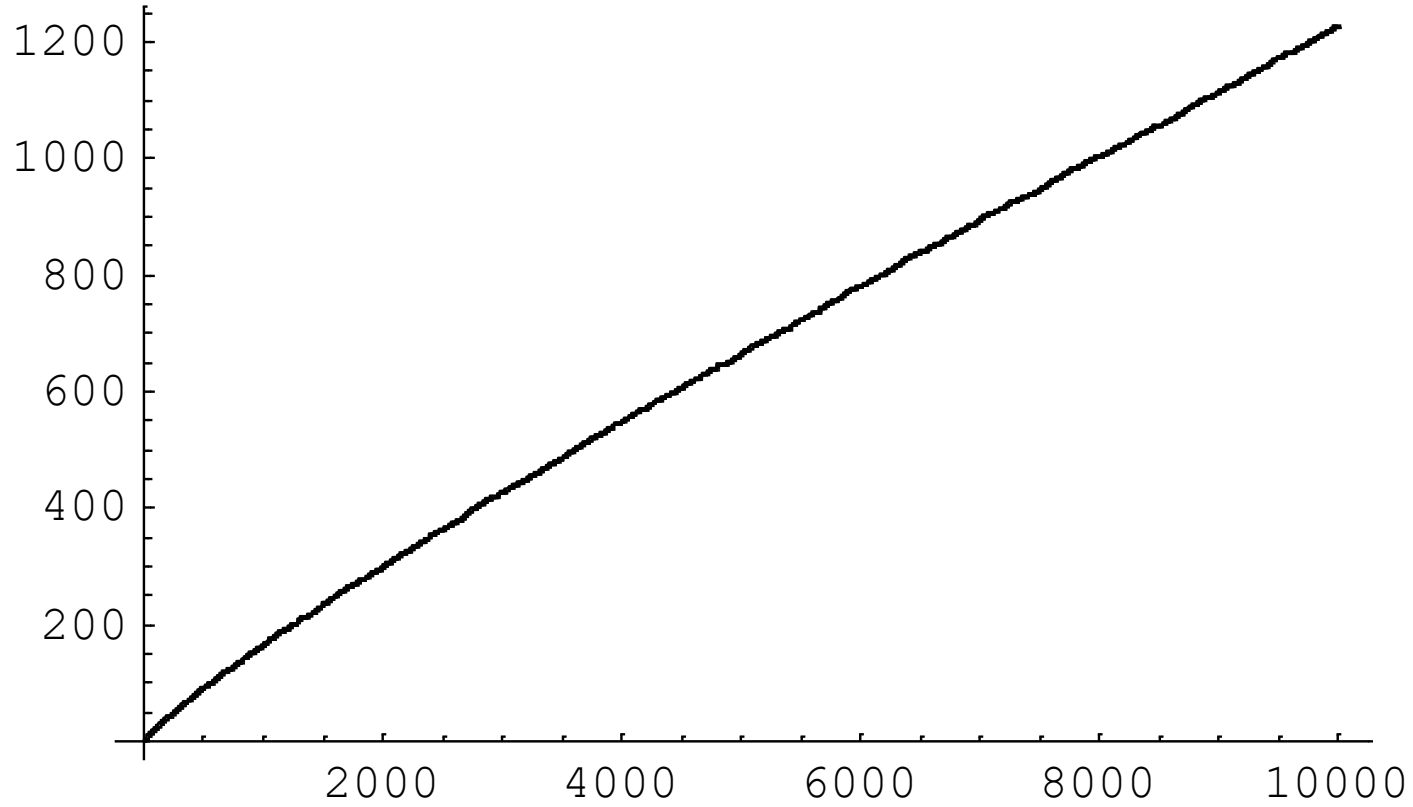
157 163 167 173 179 181 191 193 197 199

素数

Table 8.1 Primes Under 2000

2	101	211	307	401	503	601	701	809	0	1009	1103	1201	1301	1409	1511	1601	1709	1801	1901
3	103	223	311	409	509	607	709	811	911	1013	1109	1213	1303	1423	1523	1607	1721	1811	1907
5	107	227	313	419	521	613	719	821	919	1019	1117	1217	1307	1427	1531	1609	1723	1823	1913
7	109	229	317	421	523	617	727	823	929	1021	1123	1223	1319	1429	1543	1613	1733	1831	1931
11	113	233	331	431	541	619	733	827	937	1031	1129	1229	1321	1433	1549	1619	1741	1847	1933
13	127	239	337	433	547	631	739	829	941	1033	1151	1231	1327	1439	1553	1621	1747	1861	1949
17	131	241	347	439	557	641	743	839	947	1039	1153	1237	1361	1447	1559	1627	1753	1867	1951
19	137	251	349	443	563	643	751	853	953	1049	1163	1249	1367	1451	1567	1637	1759	1871	1973
23	139	257	353	449	569	647	757	857	967	1051	1171	1259	1373	1453	1571	1657	1777	1873	1979
29	149	263	359	457	571	653	761	859	971	1061	1181	1277	1381	1459	1579	1663	1783	1877	1987
31	151	269	367	461	577	659	769	863	977	1063	1187	1279	1399	1471	1583	1667	1787	1879	1999
37	157	271	373	463	587	661	773	877	983	1069	1193	1283		1481	1597	1669	1789	1889	1997
41	163	277	379	467	593	673	787	881	991	1087		1289		1483		1693			1999
43	167	281	383	479	599	677	797	883	997	1091		1291		1487		1697			
47	173	283	389	487		683		887		1093		1297		1489		1699			
53	179	293	397	491		691				1097				1493					
59	181			499										1499					
61	191																		
67	193																		
71	197																		
73	199																		
79																			
83																			
89																			
97																			

素数



素数的个数

素因子分解

□ 算数基本定理

任意整数 a > 1 都可以唯一地因子分解为

$a = p_1^{a1}p_2^{a2}\ldots p_t^{at}$ ，其中，p_i 均是素数，且$p_1<p_2<\ldots<p_t$,且每一个$a_i>0$

如. 91 = 7 x 13 ； 3600 = 24 x 32 x 52

□ 确定一个大数的素因子分解不是一件容易的事

互素和最大公因子

- **两个数 a, b 互素，如果它们没有除1以外的公因子**

如： (8, 15) = 1

- **最大公因子**

如：$300 = 2^2 \times 3^1 \times 5^2$

$18 = 2^1 \times 3^2$

因此 GCD(18, 300) = $2^1 \times 3^1 \times 5^0 = 6$

提纲

Fermat定理和Euler定理

Fermat小定理

$a^{p-1} \equiv 1 \pmod{p}$

p是素数，gcd(a, p) = 1

例题：计算2^{100}除以13的余数

$$
\begin{aligned}
2^{100} &\equiv 2^{12\times 8+4} \pmod{13} \\
&\equiv (2^{12})^8 \cdot 2^4 \pmod{13} \\
&\equiv 1^8 \cdot 16 \pmod{13} \\
&\equiv 16 \pmod{13} \\
&\equiv 3 \pmod{13}.
\end{aligned}
$$

Euler函数ø(n)

小于n且与n互素的正整数的个数

如 n = 10，{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}，{1, 3, 7, 9}； ø(10) = 4

素数 p　　ø(p) = p-1

素数p, q，有 ø(pq) = (p-1) x (q-1)

如：ø(37) = 36

ø(21) = (3–1) x (7–1) = 2 x 6 = 12

约定： ø(1) = 1

Euler函数ø(n)

口 定理

设 $n = p_1^{e1} p_2^{e2} \cdots p_r^{er}$，$p_i \neq p_j$, p_i为素数，$e_i \geq 1$,则

$$ø(n) = n (1-p_1^{-1}) (1-p_2^{-1})\cdots(1-p_r^{-1})$$

例如：$12 = 2^2 * 3$

$$ø(12) = 12 * (1-2^{-1}) * (1-3^{-1}) = 4$$

Euler函数ø(n)

Table 8.2 Some Values of Euler's Totient Function $\phi(n)$

n	$\phi(n)$	n	$\phi(n)$	n	$\phi(n)$
1	1	11	10	21	12
2	1	12	4	22	10
3	2	13	12	23	22
4	2	14	6	24	8
5	4	15	8	25	20
6	2	16	8	26	12
7	6	17	16	27	18
8	4	18	6	28	12
9	6	19	18	29	28
10	4	20	8	30	8

Euler定理

$a^{\phi(n)} \equiv 1 \pmod{n}$，对任意 a, n，gcd(a, n) = 1

另一种表示：

$a^{\phi(n)+1} \equiv a \pmod{n}$，对任意 a, n

如：

a = 3; n = 10; $\phi(10) = 4$; 则 $3^4 = 81 \equiv 1 \bmod 10$

a = 2; n = 11; $\phi(11) = 10$;则 $2^{10} = 1024 \equiv 1 \bmod 11$

Fermat小定理是Euler定理的推论，或者说， Euler定理是Fermat小定理的更一般化形式。

Euler定理

□ 与RSA有关的结果

两个素数 p 和 q， 整数m 和 n，

n = pq , 0< m <n , 则有 $m^{\phi(n)+1} = m^{(p-1)(q-1)+1} \equiv m \pmod n$

另一种表示:

$m^{k\phi(n)+1} \equiv m \pmod n$

提纲

素性测试

- **常常需要找到大的素数**
- **试除法**

例如，用小于该数平方根的所有数去试除

对较小数有效

- **基于素数性质的有选择的统计方法**

所有素数均应满足素数的性质

但某些合数（可称作伪素数）也满足素数的性质

- **确定素性的测试**

```
bool isPrime( long long n )
{
    for(long long i = 2; i*i <= n; i++)
    {
        if(n%i == 0) return false;
    }
    return true;
}
```

Miller Rabin算法

- **基于Fermat定理**
- **算法如下:**

TEST (n) is:

1. 找出整数 k, q, 其中k > 0, q 是奇数，使得 $(n–1) = 2^k q$
2. 随机选择整数 a, 1 < a < n–1
3. if $a^q \bmod n = 1$ then 返回 (“不确定");
4. for j = 0 to k – 1 do
5. if $(a^{2^j q} \bmod n = n\text{-}1)$ then 返回(“ 不确定")
6. return (“合数")

概率方面的考虑

- 如果Miller-Rabin测试返回 “合数”，则该数一定不是素数；返回“不确定”，则该数可能是素数，也可能是伪素数
- 遇到伪素数的概率 < ¼
- 用t个不同的随机选择的数a，重复做测试t次，则n是素数的概率是：$Pr = 1 - 4^{-t}$

例如：t = 10，n是素数的概率 > 0.999999

素数的分布

- **数论中的素数定理可知：n附近的素数分布情况为，平均每 ln(n)个整数中有一个素数**
- **偶数和5的倍数，都不是素数，所以只需要测试 0.4ln(n)次**

如，要找2^{200}左右的素数，则约需55次测试

- **这里只是平均意义上的结论**

有时素数分布很密，有时很松

提纲

中国剩余定理

- 在《孙子算经》中有这样一个问题：“今有物不知其数，三三数之剩二（除以3余2），五五数之剩三（除以5余3），七七数之剩二（除以7余2），问物几何？”这个问题称为“孙子问题”，该问题的一般解法国际上称为“中国剩余定理”。
- **中国剩余定理给出了以下的一元线性同余方程组有解的判定条件，并用构造法给出了在有解情况下解的具体形式**

$$(S):\quad \begin{cases} x \equiv a_1 \pmod{m_1} \\ x \equiv a_2 \pmod{m_2} \\ \quad \vdots \\ x \equiv a_n \pmod{m_n} \end{cases}$$

- 其中，$m_1, m_2, \ldots, m_n$两两互素

中国剩余定理

$$x \equiv \left(\sum_{i=1}^{k} a_i c_i \right) (\mathrm{mod}\ M)$$

$$c_i = M_i \times (M_i^{-1} \bmod m_i) \quad \text{for } 1 \leq i \leq k$$

中国剩余定理

除数 m_i	余数 a_i	最小公倍数	衍数 $M_i = M/m_i$	乘率 M_i^{-1}	c_i	各总 $a_i c_i$	答数
m_1	a_1	$M = m_1 m_2 \ldots m_k$	M_1	M_1^{-1}			$\left(\sum_{i=1}^{k} a_i c_i\right) (\mathrm{mod}\ M)$
m_2	a_2		M_2	M_2^{-1}			
…	…		…	…	…	…	
m_k	a_k		M_k	M_k^{-1}			

应用举例

- **孙子算经：今有物不知其数，三三数之剩二，五五数之剩三，七七数之剩二，问物几何?**
- **答曰二十三**

设所求物数为 X, 则

$X \equiv 2 \pmod 3$,

$X \equiv 3 \pmod 5$,

$X \equiv 2 \pmod 7$

中国剩余定理

除数 m_i	余数 a_i	最小公倍数	衍数 $M_i = M/m_i$	乘率 M_i^{-1}	c_i	各总 $a_i c_i$	答数
3	2	M= 3*5*7 =105	5*7	2	5*7*2	70*2	140+63+30=233≡23 mod 105
5	3		7*3	1	7*3*1	21*3	
7	2		3*5	1	3*5*1	15*2	

中国剩余定理

- 用于加速模运算
- 某一范围内的整数可通过它对两两互素的整数取模所得的余数来重构
- 使得非常大的数对M的模运算转化到更小的数上来进行运算

- 例如，计算

120523 = 1651 * 73 = (973 + 678) * 73 ≡？ (mod 1813)，已知1813 = 37 * 49 且（37，49）= 1

- **M = m1 * m2 = 37 * 49，其中 (37, 49) = 1**

973可用较小的两个模数37和49重构，表示为（11，42）

678可表示为（12，41）

则1651 = 973 + 678就可表示为

（ 11 + 12 mod 37, 42 + 41 mod 49）= (23, 34)

则120523 = 1651 * 73就可表示为

（23 * 73 mod 37, 34 * 73 mod 49）= (14, 32)

提纲

离散对数

本原根

- $a^{ø(n)} \bmod n = 1$
- $a^m = 1 \pmod n$, $GCD(a, n) = 1$

一定存在，因为$m = ø(n)$，（$ø(n)$ 是可能的最高指数）

m不一定最小

一旦到达m, 将会产生循环。

最小的m，成为a的阶。

- **如果一个数a的阶为ø(n)，则称a为n的本原根**

离散对数

本原根

- **若p是素数, a是p的本原根，则$a^1, a^2, a^3, \ldots, a^{p-1}$ 是模p各不相同的；**
- **并不是所有整数模n都有本原根。**

只有n是形为2，4，p^α和2 p^α的整数才有本原根，其中p是奇素数，α是正整数。

离散对数

Table 8.3 Powers of Integers, Modulo 19

a	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^{15}	a^{16}	a^{17}	a^{18}
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	4	8	16	13	7	14	9	18	17	15	11	3	6	12	5	10	1
3	9	8	5	15	7	2	6	18	16	10	11	14	4	12	17	13	1
4	16	7	9	17	11	6	5	1	4	16	7	9	17	11	6	5	1
5	6	11	17	9	7	16	4	1	5	6	11	17	9	7	16	4	1
6	17	7	4	5	11	9	16	1	6	17	7	4	5	11	9	16	1
7	11	1	7	11	1	7	11	1	7	11	1	7	11	1	7	11	1
8	7	18	11	12	1	8	7	18	11	12	1	8	7	18	11	12	1
9	5	7	6	16	11	4	17	1	9	5	7	6	16	11	4	17	1
10	5	12	6	3	11	15	17	18	9	14	7	13	16	8	4	2	1
11	7	1	11	7	1	11	7	1	11	7	1	11	7	1	11	7	1
12	11	18	7	8	1	12	11	18	7	8	1	12	11	18	7	8	1
13	17	12	4	14	11	10	16	18	6	2	7	15	5	8	9	3	1
14	6	8	17	10	7	3	4	18	5	13	11	2	9	12	16	15	1
15	16	12	9	2	11	13	5	18	4	3	7	10	17	8	6	14	1
16	9	11	5	4	7	17	6	1	16	9	11	5	4	7	17	6	1
17	4	11	16	6	7	5	9	1	17	4	11	16	6	7	5	9	1
18	1	18	1	18	1	18	1	18	1	18	1	18	1	18	1	18	1

离散对数

- **求 x，以满足 $y = g^x \pmod p$**
- **可以写作 $x = \log_g y \pmod p$**
- **如果g是p的本原根，则x一定存在；否则，不一定存在。**

例如：

$x = \log 34 \bmod 13$ 无解

$x = \log 23 \bmod 13 = 4$

- **指数运算相对容易，求离散对数问题是困难的**

离散对数

- **求 x，以满足 $y = g^x \pmod p$**
- **可以写作 $x = \log_g y \pmod p$**
- **如果g是p的本原根，则x一定存在；否则，不一定存在。**

例如：

$x = \log 34 \bmod 13$ 无解

$x = \log 23 \bmod 13 = 4$

- **指数运算相对容易，求离散对数问题是困难的**

离散对数

定义

- **若m > 1, (a, m) = 1, 则使得同余式 $a^i \equiv 1 (\mathrm{mod}\ m)$成立的最小正整数i，叫做a对模m的离散对数。**
- **指数一定是欧拉函数的因子**
- **对任意整数b和模数p的本原根a，有唯一的幂i，使得$b \equiv a^i \bmod p$, 其中$0 \le i \le p-1$**

该指数i称为以a为底模p的离散对数，记为 $\mathrm{dlog}_{a,\ p}(b)$

- **离散对数不仅与模有关，而且与本原根有关。**

例如：2对模7的指数是3，对模11的指数是10，所以，2是模11的一个本原根，而不是模7的本原根；

$\mathrm{dlog}_{2,\ 9}(8) = 3$

离散对数

Table 8.4 Tables of Discrete Logarithms, Modulo 19

(a) Discrete logarithms to the base 2, modulo 19

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$\log_{2,19}(a)$	18	1	13	2	16	14	6	3	8	17	12	15	5	7	11	4	10	9

(b) Discrete logarithms to the base 3, modulo 19

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$\log_{3,19}(a)$	18	7	1	14	4	8	6	3	2	11	12	15	17	13	5	10	16	9

(c) Discrete logarithms to the base 10, modulo 19

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$\log_{10,19}(a)$	18	17	5	16	2	4	12	15	10	1	6	3	13	11	7	14	8	9

(d) Discrete logarithms to the base 13, modulo 19

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$\log_{13,19}(a)$	18	11	17	4	14	10	12	15	16	7	6	3	1	5	13	8	2	9

(e) Discrete logarithms to the base 14, modulo 19

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$\log_{14,19}(a)$	18	13	7	8	10	2	6	3	14	5	12	15	11	1	17	16	4	9

(f) Discrete logarithms to the base 15, modulo 19

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$\log_{15,19}(a)$	18	5	11	10	8	16	12	15	4	13	6	3	7	17	1	2	14	9

作业

1、描述并用代码实现欧几里得算法。

2、描述并用代码实现扩展欧几里得算法。

3、描述并用代码实现中国剩余定理求解过程（n取3）。

本章结束

~End~

是什麼讓火焰燃燒，是叢柴之間的空隙，它們靠此呼吸。

What makes a fire burn is space between the logs, a breathing space.