素数：

- 一个大于 1，且只能被 1 和它本身整除的正整数，称为素数；否则称为合数.
- 由此可知，正整数集合可分为三部分: 素数、合数 和 1
- 一些性质

  素数的个数是无穷的

  除 2 以外的素数一定是奇数，也称为奇素数

  任意两个相邻的正整数 n 和 n + l(n > 3)中必有一个不是素数

  相邻整数均为素数的只有 2 和 3

素因子：

若 b|a，且 b 是素数，称 b 是 a 的素因子

例：

12=3×4, 3 是 12 的素因子, 2 也是 12 的素因子，而 4 不是（3|12，2|12）

整数分解的唯一性定理

定理：

任意一个正整数 a>1 总可以分解为一系列素数乘积的形式，而且分解形式是唯一的

$a = p_1^{a1} \times p_2^{a2} \times p_3^{a3} \times \cdots\cdots \times p_n^{an}$

例：

$36 = 2^2 \times 3^2$

$3600 = 2^4 \times 3^2 \times 5^2$

Q 如何生成素数?

现代密码学，特别是公钥密码学，常用随机的大素数,习惯上，常用符号 p 或 q 表示素数

Q: 如何生成一个随机的大素数?

先生成一个随机数，然后将之分解得到其素因子，从而得到素数，这种方法是否可行？

不行！

就目前而言，对某些大整数进行素因子分解是计算上不可行的

生成素数的正确方法 —— 素性检测：随机产生一个大奇数，然后检测它是否是素数

在诸多素性检测算法中，Miller-Rabin 算法 是容易实现且广泛使用的算法

随机素数的生成过程

① 随机产生一个 n 比特的随机数 p (如通过伪随机序列发生器)。

② 将 p 的最高位和最低位设置为 1 (最高位设置为 1 的目的是确保素数达到最大有效长度，最低位设置为 1 的目的是确保该数是奇数)。

③ 检查 p 不能被所有小于 2000 的素数整除（有方法可使这一步做得很快)。

④ 随机选择 a，且 a<p。

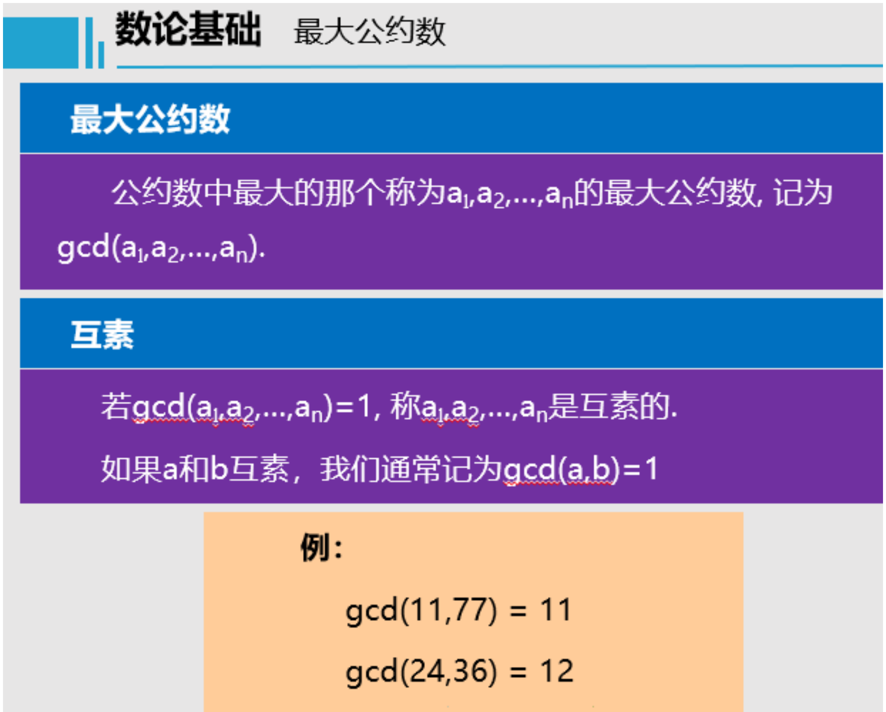
⑤ 用 a 对 p 进行素性测试(如用 Miller-Rabin 算法)。若 p 没有通过测试,抛弃 p,转到 ①（或将 p 加 2 作为新的 p,然后转到③)。

⑥ 如果通过测试,转到④。如果 p 通过足够次数的测试(如 5 次)，认为 p 是素数。

素性检测算法是概率算法，会有一定的错误概率

- 它永远不会把素数误判为合数，却可能把合数误判为素数，但概率很低
- 对一个随机数重复进行多次素性测试，误判的概率会比你中六合彩还低
- 因此，误判这个问题我们根本不必担心

公约数(公因子)：设 $a_1,a_2,\cdots,a_n$ 和 d 都是正整数($n\geq2$). 若 $d|a_i$ ($1\leq i\leq n$)，则称 d 是 $a_1,a_2,\cdots,a_n$ 的公约数(公因子).



公约数中最大的那个称为 $a_1,a_2,\cdots,a_n$ 的最大公约数，记为 $\gcd(a_1,a_2,\cdots,a_n)$.

最大公约数的性质

- 在互素的正整数中，不一定有素数.

例：gcd(25,36)=1，但 25 和 36 都不是素数.

- 在个数不少于 3 个的互素正整数中，不一定两两互素.

例: gcd(6,10,15)= 1,

但 gcd(6,10)=2,

gcd(6,15)=3,

gcd(10,15)=5.

数论基础 最大公约数

- **基于的原理**

若 a = b*d + r，则 gcd(a,b) = gcd(b,r)。

(a是被除数，b是除数，d为商,r为余数)

例：

a=38, b=26, 则gcd(a,b)=2

欧几里得算法计算过程：

① 38=26*1+12

② 26=12*2+2

③ 12=2*6

最后一个非零余数(最后一步的除数)就是所求的gcd

数论基础 最大公约数

重要定理

设两个正整数a、b, 最大公约数设为gcd(a,b)，则必存在两个整数 x 和 y,使得

$$ax + by = gcd(a,b)$$

推论

当a和b互素时，必存在两个整数 x 和 y，使得

$$ax + by = 1$$

- 该推论在后面介绍公钥密码学时很有用

如何计算该定理中的x和y?

- 利用欧几里得算法计算a和b的最大公约数时，一些中间结果可以用于计算x和y
- 因此，将欧几里得算法改版一下便可。所得新算法称为“扩展的欧几里得算法”

Q: 如何计算a和b的最小公倍数?

$$\mathrm{lcm}(a,b) = ab / \gcd(a,b)$$

存在的问题：计算大量乘法和除法，效率仍然不够高

数论基础　同余

- **定义**

  若 a mod n = b mod n = r，则记

  $a \equiv b \pmod{n}$

  称a和b在模n下是同余的　**(a和b在模n下有相同的余数)**

例：

$13 \equiv 20 \pmod{7}$

$20 \equiv 7 \pmod{13}$

数论基础　模运算的性质

模运算的性质

- $(a+b) \bmod n = (a \bmod n + b \bmod n) \bmod n$
- $(a-b) \bmod n = (a \bmod n - b \bmod n) \bmod n$
- $(a \times b) \bmod n = (a \bmod n \times b \bmod n) \bmod n$
- 其他性质
  - 交换律
  - 结合律
  - 分配律

乘法链运算：

Q 如何计算 a^m mod n？

A 将 m 化为二进制数

数论基础 乘法链算法

- 计算a^{25} mod n的过程：m=25是11001

u=m	s=1	t=a
u=11001	s=(st) % n =a % n	t=t^2 % n =a^2 % n
u=11001		t=t^2 % n =a^4 % n
u=11001		t=t^2 % n =a^8 % n
u=11001	s=(st) % n = a^9 % n	t=t^2 % n =a^{16} % n
u=11001	s=(st) % n = a^{25} % n	

```
unsigned long qe2  (unsigned long a, unsigned long m, unsigned long n)
{
    unsigned long s=1,t=a,u=m;
    while(u)
    {
        if(u & 1) s = (s * t) % n;
        u >>=1;
        t = (t * t) % n;
    }
    return s;
}
```

时间复杂度：O(logm)

30

欧拉函数的计算结果：小于等于 n 且与 n 互素的正整数的个数。

欧拉函数的性质

- **性质**：若 p、q 都是素数，则 $\varphi(p) = p - 1$ ， $\varphi(p\times q) = \varphi(p)\varphi(q) = (p - 1)(q - 1)$

- **定理**：若 p 为素数，k 为正整数，则 $\varphi(p^k) = p^{k-1}(p - 1)$

- **推论**：

  若 gcd(a,b) = 1，则 $\varphi(ab) = \varphi(a)\varphi(b)$

- **欧拉定理** 设 n≥2，如果 gcd(a,n)=1，则 $a^{\varphi(n)} \equiv 1 \pmod n$，

同理有 $a^{\varphi(n)+1} \equiv a \pmod n$

容易求得 $\varphi(60)=16$

Q(60)=Q(4)xQ(15)=2xQ(3)*Q(5)=2x2x4;

数论基础 欧拉定理的应用

- 求 $13^{2001} \bmod 60$
- **解:** 因为gcd(13，60) = 1，则可用欧拉定理计算，故而有

$$13^{\phi(60)} \equiv 1 \pmod{60}$$

容易求得 $\phi(60) = 16$

故而，$13^{16} \equiv 1 \pmod{60}$

所以，$13^{2001} = 13^{16\times125+1} \equiv 13 \pmod{60}$

- **费马小定理：** 若 p 是素数, gcd(a,p)=1,则 $a^{p-1} \equiv 1 \pmod p$同理有 $a^{p} \equiv a \pmod p$

例:

$$2^{5-1} \equiv 1 \pmod 5$$

$$4^{11-1} \equiv 1 \pmod{11}$$

数论基础 费马小定理的应用

- 求$310^{198} \bmod 199$。
- 解:

因为199是素数, 且gcd(310,199)=1，根据费马小定理，有

$$310^{199-1} \equiv 1 \pmod{199}$$

所以, $310^{198} \equiv 1 \pmod{199}$

二次剩余：

- **定义：**

设 a 是小于 n(n>1)的正整数，且 gcd(a,n)=1。如果存在 x，使

$$x^2 \equiv a \pmod n$$

成立，那么称 a 是模 n 下的二次剩余。

二次剩余的特性

- **性质 1：**二次剩余的判定标准
  如果 $a^{(p-1)/2} \equiv 1 \pmod p$，其中 p>2 是素数，1≤a<p
  则 $a \in QR_p$
- **性质 2：**当模数是素数时，设为 p>2
  - 模 p 下的二次剩余恰有(p-1)/2 个，与其二次非剩余的数目相同(各占一半)
  - 如果 $a=x^2 \pmod p$是二次剩余，那么 a 恰好有两个平方根。一个是 x,另一个是 p-x。

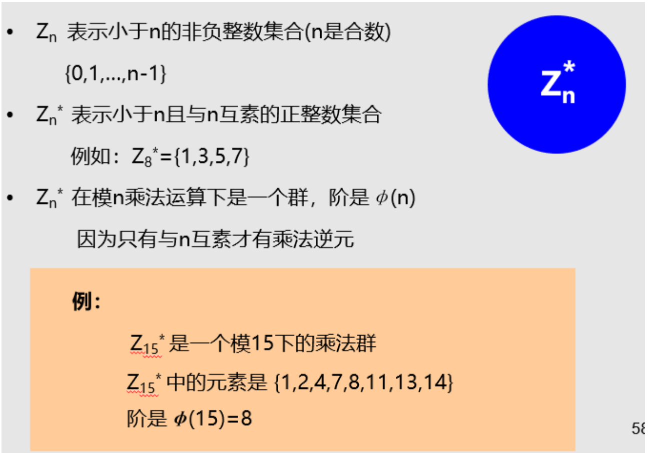
群，循环群的几个性质

- 任何循环群必是交换群
- 设有限循环群的阶为 n，则
  ① 生成元的阶也是 n，其他元素的阶必是 n 的因子
  ② 生成元的个数为$\varphi(n)$
    - 例循环群中有 15 个元素，那么生成元的个数为$\varphi(15)=\varphi(3)\varphi(5)=2*4=8$
  ① 阶为 m 的元素的个数共有$\varphi(m)$个

密码学中的常用群

- Z_p 表示小于素数 p 的非负整数集合
{0,1,2,⋯,p-1}
- Z_p^* 表示小于素数 p 且与 p 互素的正整数集合
{1,2,⋯,p-1}
- Z_p^* 在模 p 乘法运算下是一个循环群
- **相关性质：**
  - 单位元是 1

- 群的阶是 p-1，生成元的个数是φ(p-1)
    - 例：Z_7^*是一个循环群，生成元共有$\varphi(6)=2$ 个，分别是 3 和 5。



域　有限域

- 若域 F 是有限的，称 F 为有限域，又称伽罗瓦域
- **定理 1**
    - 有限域的元素个数必为 p^n 的形式(p 为素数)，记为
- **定理 2**
    - 有限域的乘法群是循环群
- **定理 3**
    - 同阶(元素数目相同)的有限域都是同构的
- 密码学中，我们关注 F_p(又称素数域)和

- **信息论告诉我们**
    - 除一次一密以外，任何密码都是可以破解的。
- **计算复杂性理论告诉我们**
    - 在宇宙爆炸前，它们是否可以被破解。而破解它们所花费的时间和空间是否已超出你所能承受的底线。

现代密码学将安全性构建在计算复杂性理论之上

- 公钥密码学以目前的计算方法不能有效解决的问题为构造基础，这些问题被称为“困难问题”。
- 密码体制的安全性就在于困难问题的困难性

计算复杂性 问题复杂性分类

- **问题本身有着内在固有的复杂性**，这与解决问题的算法的复杂度是不同的。
- 计算复杂性理论研究的主要内容：对问题的复杂性进行分类。
- 所用的工具便是图灵机

计算复杂性 P 问题

- **P 问题**
  - 存在一个确定性图灵机，可以在多项式时间内解决的问题
- 通常认为，多项式时间算法是有效算法，多项式时间内可解决的问题是容易问题（**P 问题是容易问题**)
  - 但这种想法是不精确的。当 k 很大时，例如 k=200，即使 n 很小，例如 n=2,3，$O(n^{200})$也非常大
  - 但多项式时间算法毕竟比指数和超多项式时间算法快得多，因此这种说法是可以接受的

计算复杂性 NP 问题

- **NP 问题**
  - 存在一个非确定性图灵机，可以在多项式时间内解决的问题。
- 也就是说，NP 问题用非确定性图灵机很容易解决。
- 因为在确定性图灵机上未发现多项式时间的算法，通常认为 NP 问题是难以解决的，俗称 **困难问题**

- **P 问题也属于 NP 问题的范畴**
  - 因为确定性图灵机上在多项式时间内可以解决的问题，在非确定性图灵机上用多项式时间也可以解决。（把猜测阶段省略掉即可）
- **但我们一般所说的困难问题或 NP 问题，通常不包括 P 问题。**

计算复杂性

计算复杂性 NPC 问题

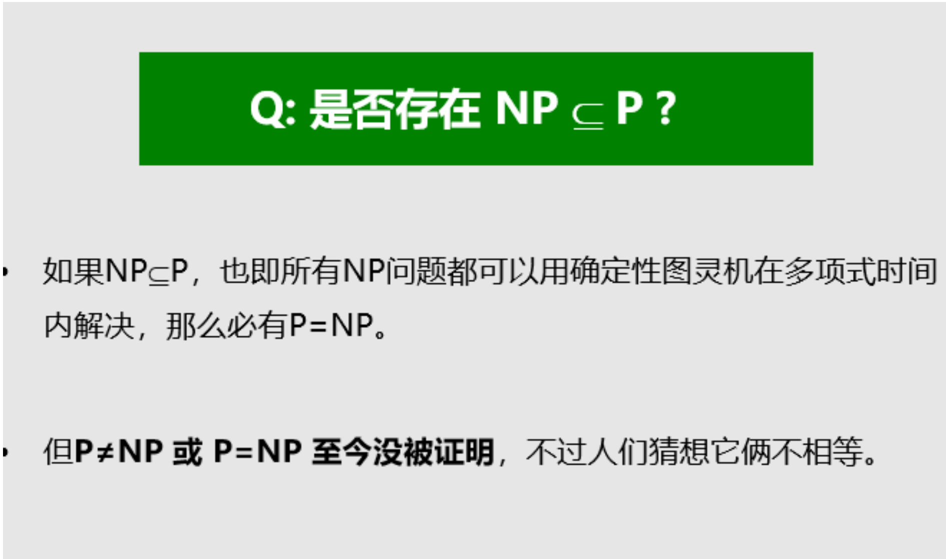
- NP 中有一些特殊问题，如果找到一个确定的多项式时间算法可以解决这些问题中的一个，解 NP 中的任何问题都能找到多项式时间算法。
- 这类问题被称作 NP 完全问题，记为 **NPC**.

计算复杂性：P 与 NP 的关系

- 由前述可知

$P \subseteq NP$

- 任何在确定性图灵机上在多项式时间内可以解决的问题，在非确定性图灵机上在多项式时间内同样可以解决



“P 是否等于 NP” 是计算复杂性理论未解决的中心问题