中国科学技术大学 计算机科学与技术系
University of Science and Technology of China
DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY

数 论 算 法

Number-Theoretic Algorithms

主讲人　徐 云

Fall 2018, USTC

数论算法（Ch31）

1 初等数论记号（31.1）

2 最大公约数（31.2）

3 模运算（31.3）

4 求解线性模方程（31.4）

5 中国余数定理（31.5）

6 RSA公钥密码系统（31.7）

7 素数判定（31.8）

初等数论记号（31.1）

- 算术计算中的输入规模和成本
- 除定理
- 公约数和最大公约数
- 互素数和最大公约数

算术计算中的输入规模和成本

- **Def.1**

  一个输入整数$a_1,a_2,\ldots,a_k$的算法是多项式时间算法， 如果其运行时间是以$\log a_1,\log a_2,\ldots,\log a_k$为多项式时间的， 即以输入参数二进制编码长度的多项式时间的。

- 如， 两个β位的整数乘法， 普通乘法使用$\Theta(\beta^2)$,

  改进的算法使用$\Theta(\beta^{\log 3})$， 甚至$\Theta(\beta\log\beta\log\log\beta)$

除定理

- **概念**：可除性， 素数， 合数

- **Th31.1**

  对任何整数a和正整数n，存在唯一整数q和r，使得

  $a=qn+r$，这里$q=\lfloor a/n \rfloor$，$0\leq r < n$。

  注：q为商，r为余数

- Def. 2

  模n的等价类：$[a]_n=\{ a+kn \mid k\in Z\}$

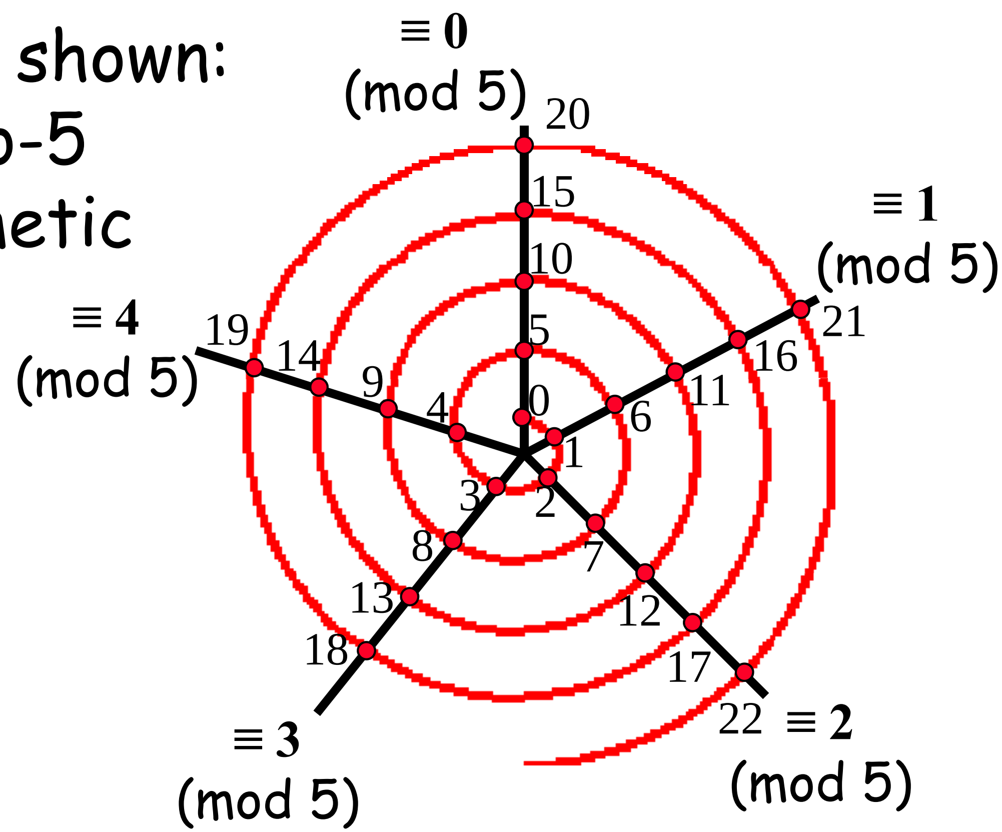
  - 性质：

    如果$a\in[b]_n$，则$a\equiv b \pmod n$

除定理

- **Spiral Visualization of mod:**

Example shown:
modulo-5
arithmetic



公约数和最大公约数

- 概念：约数，公约数，最大公约数

s是ax+by中最小正整数
1、gcd(a,b) | s ==> gcd(a,b) <= s
2、下证s | gcd(a,b)
a = qs + (a mod s), 0 <= a mod s < s (#)
= q(ax+by) + (a mod s)
==> a mod s = (1 - qx)a - by (*),即该余数表示成a和b的线性组合
由(#),(*) ==> a mod s = 0 ==> s | a,同样可得s | b ==> s | gcd(a,b)

- **Th31.2**

  a，b为不全为零的两个整数，则最大公约数gcd(a, b)是 $\{ax+by \mid x,y\in Z\}$中最小的正整数。

  -系1: 对所有整数a和b，如果$d|a$，$d|b$，则$d|gcd(a, b)$

  -系2: 对所有整数a和b，和非负整数n，有

  $gcd(an, bn) = n\,gcd(a, b)$

  -系3: 对所有正整数n、a和b，如果$n|ab$且$gcd(a, n)=1$，则 $n|b$

互素数和素数唯一性分解定理

- **Def.3**：

  如果$gcd(a, b)=1$，称整数a和b为互素数

- **Th31.6**

  $\forall a, b, p \in Z$，如果$gcd(a, p)=1$和$gcd(b, p)=1$，则 $gcd(ab, p)=1$

- **Th31.8(唯一分解定理)**

  一个合数a能被唯一写成形式 $a=p_1^{e1}p_2^{e2}...p_r^{er}$

  这里p_i是素数，$p_1 < p_2<...<p_r$，e_i是正整数

  如，$6000=2^4\times 3\times 5^3$

数论算法（Ch31）

1 初等数论记号（31.1）

2 最大公约数（31.2）

3 模运算（31.3）

4 求解线性模方程（31.4）

5 中国余数定理（31.5）

6 RSA公钥密码系统（31.7）

7 素数判定（31.8）

最大公约数（31.2）

- 一种直观的解GCD
- Euclid's Algriothm
- Euclid's Algriothm的运行时间
- 扩展的Euclid's Algriothm

一种直观解GCD

$a = p_1^{e1}p_2^{e2}\ldots p_r^{er}$ 和 $b = p_1^{f1}p_2^{f2}\ldots p_r^{fr}$

$\Rightarrow \gcd(a, b) = p_1^{\min(e1, f1)}p_2^{\min(e2, f2)}\ldots p_r^{\min(er, fr)}$

注: 这种解法需要整数的素因子分解，而素因子分解是一个很难的问题(NP难问题)

Euclid's Algorithm

- **Th.31.9(GCD Recursion)**

  对任何非负整数a和正整数b， 有gcd(a, b) = gcd(b, a mod b)

- **Euclid's Algorithm**

```
Euclid(a, b)
{  if b=0 then
        return a;
   else
        return Euclid(b, a mod b);
}
```

- 例：gcd(30, 21)

Euclid's Algorithm的运行时间

- **Th.31.11(拉姆定理)**

  对整数$k \geq 1$，如果$a > b \geq 1$且$b < F_{k+1}$，

  则Euclid(a, b)递归调用的次数小于k。

  注：

  - Euclid's Alg.递归调用的次数为$O(\log b)$
  - 算法应用到二个β位整数上，除法的时间复杂度是$O(\beta^2)$

    算法耗费$O(\beta)$算术运算和

    $O(\beta^3)$位运算

Gabriel Lamé
1795-1870

扩展的Euclid's Algorithm

- 问题

  Find x, y s.t. gcd(a, b) =ax+by

- 算法

```
ExtendedEuclid(a, b)
{ if b=0 then
      return (a, 1, 0);
   (d', x', y') ← ExtendedEuclid(b, a mod b);
   (d, x, y) ← (d', y', x'-⌊a/b⌋y');
   return (d, x, y);
}
```

- 例：把3=gcd(99, 78)表示成99和78的线性组合

例	a	b	[a/b]	d	x	y
	99	78	1	3	-11	14
	78	21	3	3	3	-1
	21	15	1	3	-2	3
	15	6	2	3	1	-2
	6	3	2	3	0	1
	3	0	/	3	1	0

x,y表示本次，x',y'表示下次的

b = a mod b

x = y'

y = x' - [a/b]y'

数论算法（Ch31）

1 初等数论记号（31.1）

2 最大公约数（31.2）

3 模运算（31.3）

4 求解线性模方程（31.4）

5 中国余数定理（31.5）

6 RSA公钥密码系统（31.7）

7 素数判定（31.8）

模运算（31.3）

- 模n的加法群和乘法群
- Euclid's phi Function
- Lagrange's Theorem

模n的加法群和乘法群

- **Def. 1:**

  设Z_n={ $[a]_n$ | 0≤ a ≤n-1}，定义加法运算$+_n$:

  $[a]_n +_n [b]_n = [a+b]_n$,

  则$(Z_n, +_n)$是有限Abelian群

  如，$(Z_6, +_n)$，Z_6 ={$[0]_6$, $[1]_6$, $[2]_6$, $[3]_6$, $[4]_6$, $[5]_6$}

- **Def. 2:**

  互素

  设Z_n^*={ $[a]_n \in Z_n$ | gcd(a, n)=1 }，定义乘法运算$\times_n$:

  $[a]_n \times_n [b]_n = [a \times b]_n$,

  则$(Z_n^*, \times_n)$是有限Abelian群

  如，$(Z_{15}^*, \times_n)$，Z_{15}^* ={1, 2, 4, 7, 8, 11, 13, 14}

Euclid's phi Function

- **Theorem(Euclid's phi Func.)**

  令φ(n)为Z_n^*的size，则有$\varphi(n) = n\prod_{p|n}(1-\frac{1}{p})$

  能整除n的素数集

- **例：**

  $Z_{15}^* = \{1, 2, 4, 7, 8, 11, 13, 14\}$

  $|Z_{15}^*| = 15(1-1/3)(1-1/5) = 8$

  15 = 3×5

Lagrange's Theorem

- **Th. 31.15 (Lagrange's Theorem)**

  如果H是有限群G的子群，那么|H|整除|G|

  系: $|H| \le |G|/2$

数论算法（Ch31）

1 初等数论记号（31.1）
2 最大公约数（31.2）
3 模运算（31.3）
4 求解线性模方程（31.4）
5 中国余数定理（31.5）
6 RSA公钥密码系统（31.7）
7 素数判定（31.8）

求解线性模方程（31.4）

- 问题描述
- <a>群表示和构造定理
- 求解方法

问题描述

求x满足下面模方程:

$$ax \equiv b \pmod{n}$$

这里 $a > 0$, $n > 0$

<a>群表示和构造定理

- **Def.:**

令<a>是Z_n上由a生成的子群，$\langle a\rangle=\{\, ax \bmod n \mid x\geq 0, x\in Z\}$

- **性质:**

方程$ax \equiv b \pmod{n}$有解 $\overset{\text{充分必要}}{\Leftrightarrow}$ $b\in\langle a\rangle$

- **Th31.20(构造定理)**

对正整数a和n，d=gcd(a, n)，则有

$\langle a\rangle = \langle d\rangle = \{\, 0, d, 2d, \ldots, (n/d - 1)d \,\}$

为Z_n上子群，且$|\langle a\rangle| = n/d$

系13.21: $ax \equiv b \pmod{n}$有解 $\overset{\text{充分必要}}{\Leftrightarrow}$ $\gcd(a, n) \mid b$

<a>群表示和构造定理

- 例：判断$4x \equiv 2 \pmod 6$ 和 $4x \equiv 3 \pmod 6$有无解

$\because \gcd(4, 6) \mid 2 \quad \therefore 4x \equiv 2 \pmod 6$有解

$\because \gcd(4, 6)$不整除3 $\quad \therefore 4x \equiv 3 \pmod 6$无解

注：

	i:	0	1	2	3	4	5
<4>:	4i mod 6 =	0	4	2	0	4	2

可以看出：第一个方程有二个解2，5；

两种解：2+6i，5+6i

第二个方程无解

求解方法

- 先求特解
  - Th31.23 设$d=gcd(a, n)$且$d=ax'+ny'$，对某个整数x'和y'。
    如果$d|b$，则有特解$x_0 = x'(b/d) \bmod n$

  证：

$$
\begin{aligned}
ax_0 &\equiv ax'(b/d) \pmod{n} \\
&\equiv d(b/d) \pmod{n} \quad // \because ax'+ny'=d \quad \therefore ax'\equiv d \bmod n \\
&\equiv b \pmod{n}
\end{aligned}
$$

- 求全部解 构造定理
  - Th31.24 设x_0为$ax \equiv b \pmod{n}$的一个解，则有d个不同解：

$$x_i \equiv x_0 + i(n/d) \quad i=0,1, \dots , d-1$$

  - 算法：ModularLinearEquationSolver(a, b, n)

求解方法

- 示例：$14x \equiv 30 \pmod{100}$

解:

①调用ExtendedEuclid(14, 100) $\Rightarrow$ $(d, x', y')=(2, -7, 1)$

② $\because$ $2|30$， $\therefore$特解$x_0 = -7\times30/2 \pmod{100} = 95$

③调用 求全部解算法：二个解95， 45

$x_0 = 95 + 0\times100/2 = 95$

$x_1 = 95 + 1\times100/2 = 145 \equiv 45 \pmod{100}$

数论算法（Ch31）

1 初等数论记号（31.1）

2 最大公约数（31.2）

3 模运算（31.3）

4 求解线性模方程（31.4）

5 中国余数定理（31.5）

6 RSA公钥密码系统（31.7）

7 素数判定（31.8）

中国余数定理（31.5）

- 一个中国古代问题
- a模n的逆存在唯一性定理
- 中国余数定理

一个中国古代问题

- **问题**

  某物不知其数,

  三分之余二,

  五分之余三,

  七分之余二,

  此物几何？

- 该问题可以表示成线性同余方程组为

$$\begin{cases} x \equiv 2 \pmod{3} \\ x \equiv 3 \pmod{5} \\ x \equiv 2 \pmod{7} \end{cases}$$

a模n的逆存在唯一性定理

- **Def.：**

  若x使得 $xa \equiv 1 \pmod n$，称x为a模n的逆

- **Theorem:** 证明见笔记

  若a和n互素，n>1，则唯一存在a模n的逆。

  注：

  - 以下记a模n的逆为a^{-1}，使$a^{-1}a \equiv 1 \pmod n$
  - 本定理实际上给出了求a模n的逆的方法

- 例：(1)求3模7的逆；　(2)求$3x \equiv 4 \pmod 7$

中国余数定理

- **CRT(China Remainder Theorem)** 证明见笔记

  令$n_1, n_2, \ldots, n_k$为两两互素的正整数，则同余方程组

$$x \equiv a_1 \pmod{n_1}$$
$$x \equiv a_2 \pmod{n_2}$$
$$\ldots$$
$$x \equiv a_k \pmod{n_k}$$

  有唯一的模$n = n_1 n_2 \ldots n_k$解x。

 注：本定理实际上给出了求解方法

- **例(一个中国古代问题)**

数论算法（原书Ch31）

1 初等数论记号（31.1）

2 最大公约数（31.2）

3 模运算（31.3）

4 求解线性模方程（31.4）

5 中国余数定理（31.5）

6 RSA公钥密码系统（31.7）

7 素数判定（31.8）

RSA公钥密码系统（31.7）

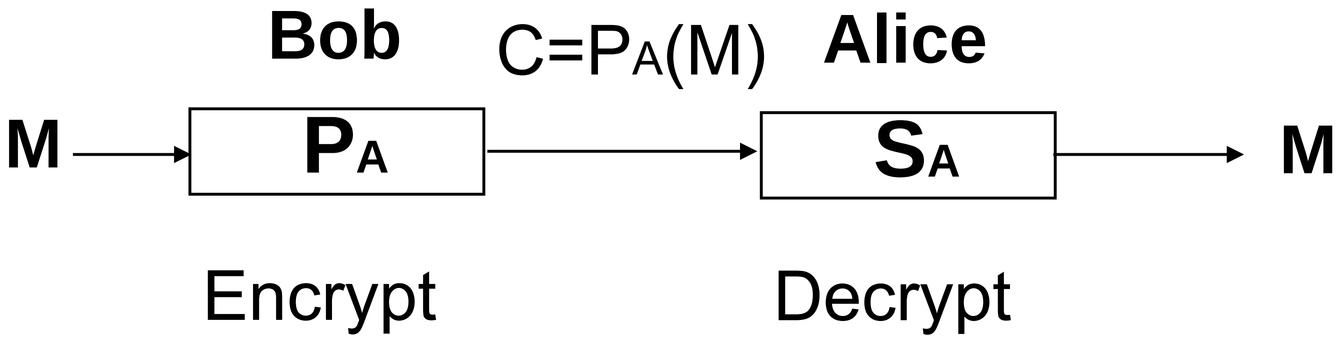
- 公钥密码系统
- RSA系统
- RSA的正确性

公钥密码系统

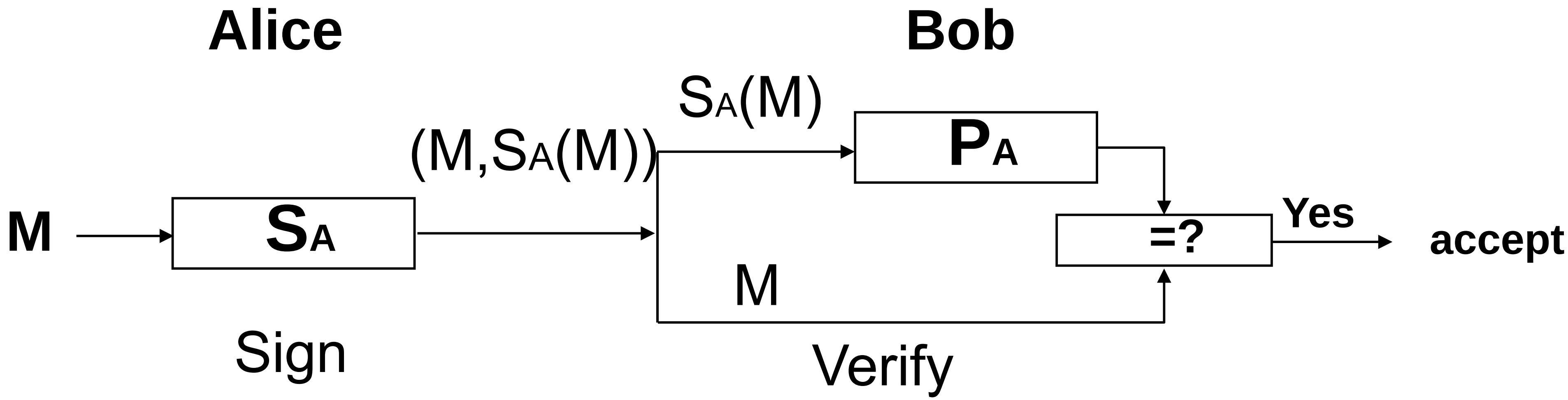
- **特征：**

  公开加密密钥，而解密密钥保密。

- 公钥系统加解密过程



- 数字签名过程



RSA系统

- **构造过程**

  ①随机选择两个大素数p和q， p<>q 二者不相等

  ②计算n=pq

  ③选择小的奇整数e， 使得e与(p-1)(q-1)互素

  ④计算e模(p-1)(q-1)的逆d， 即$ed \equiv 1 \pmod{(p-1)(q-1)}$

  ⑤公布e,n为RSA的公钥 大整数为公钥

  ⑥保存d,p,q为RSA的私钥

- **加解密过程**

  - 加密: $P(M)=M^e \bmod n$ (=C)
  - 解密: $S(C)=C^d \bmod n$

- 示例：Alice发送信息“9726”给Bob 例见笔记

RSA的正确性

- **Fermat定理**

  如果p是素数， 则$a^{p-1} \equiv 1 \pmod{p}$

- **Th31.36(Correctness of RSA)**

  RSA算法是正确的

数论算法（Ch31）

1 初等数论记号（31.1）

2 最大公约数（31.2）

3 模运算（31.3）

4 求解线性模方程（31.4）

5 中国余数定理（31.5）

6 RSA公钥密码系统（31.7）

7 素数判定（31.8）

素数判定（31.8）

- 素数的分布
- 简单的素数测试算法
- 伪素数测试算法
- Miller-Rabin随机算法

素数的分布

- **Def.：**

令π(n)表示小于或等于n的素数个数

如， π(15) = 6， 因为2， 3， 5， 7， 11， 13

- **Theorem 31.37**(Prime number theorem):

$$\lim_{n\to\infty}\frac{\pi(n)}{n/\ln n}=1$$

注: 随机选择一个整数是素数的概率为1/ln*n*

简单的素数测试算法

- **Theorem:**

  如果n被某个整数2,3,…,$\lfloor n^{(1/2)} \rfloor$整除，则n不是素数。

  注：

  - 该算法最坏时间是$\theta(n^{(1/2)})$
  - 如果n是βbit，即$\beta = \lfloor \log n \rfloor + 1$，则$n^{(1/2)} = \theta(2^{(\beta/2)})$

伪素数测试算法

- **Def.：**

  n称为基a的伪素数，如果n是合数且满足

  $$a^{(n-1)} \equiv 1 \pmod{n} \qquad (31.8.1)$$

  注：

  - 由Fermat定理知：素数一定满足上式，但满足上式的n未必一定是素数。

    如，341有 $2^{(340)} \equiv 1 \pmod{341}$，而$341=11\times 31$
  - 伪素数比较稀少：10,000以内仅有22个，前4个为 341，561，645，1105

- 系：如果存在a，使得(31.8.1)式不成立，则n一定是合数

伪素数测试算法

- 伪素数测试算法

```
PseudoPrime(n)
{   if ModularExponentiation(2,n-1,n) !≡ 1 (mod n) then
        return Composite;
    else
        return Prime;
}
```

注:

- 该算法判断出合数总是正确的;
- 如果算法给出的结论是素数，仅在基2的伪素数情形出错。

伪素数测试算法

- **Def.：**

  如果对所有的$a \in Z_n^*$，合数n满足(31.8.1)式，则称n为Carmichael数。

注：

- Carmichael数非常稀少，小于100,000,000的Car数仅有255个，前3个数为561，1105，1729；
- 但这类数有无穷多个。
- 说明存在Car伪素数，对任何基a都使得(31.8.1)式成立，这样造成伪素数测试算法的失败。

Miller-Rabin随机算法

- 伪素数测试算法的缺陷
- MR算法的改进措施
- MR随机算法
- 计算示例
- 算法的复杂性和误判率
- MR v.s. Pseudo Prime

伪素数测试算法的缺陷

- 仅选择一个基进行测试

  如，对a=2，用条件$a^{(n-1)} \equiv 1 \pmod{n}$来判断。

  存在的缺陷：

  - 尽管出错率很小，但与n相关。

    n若是随机选取的512位二进制数，出错率为＜$1/10^{20}$；

    n若是随机选取的1024位二进制数，出错率为＜$1/10^{41}$。

- 选择所有的基进行测试

  即，对所有$a \in \{2,3,\ldots,n-1\}$。

  存在的缺陷：

  - 工作量很大，但并不能排除Carmichael数。

MR算法的改进措施

- **措施1：随机选择若干个基，以减少计算量**

  即，随机选择s个基a，$a \in \{2,3,\ldots,n-1\}$。进行Fermat定理测试。

- **措施2：在计算a^(n-1) mod n过程中，加入合数证人(Witness)的查找**
  - 即查找是否存在模n余1的非平凡平方根
    (依据P560系31.35)

- **系31.35**

  如果存在模n余1的非平凡平方根，则n是合数

MR随机算法

Alg.由MR和Witness二个例程组成,

设$n-1=2^t u$, 这里$t\ge 1$, u是奇整数

$\therefore\ a^{n-1} \equiv (a^u)^{2^t} \pmod n$

- **Witness(a, n)**

```
{   x0 ← ModularExponentiation(a,u,n);
    for i ← 1 to t do
    {   xi ← (x(i-1)^2 mod n);
        if xi=1 and x(i-1)<>1 and x(i-1)<>n-1 then return True;
    }
    if xt<>1 then return True;
    return False;
}
```

$x_0 \leftarrow$ ModularExponentiation(a,u,n);

for $i \leftarrow 1$ to t do

$x_i \leftarrow (x_{i-1}^2 \bmod n)$;

if $x_i=1$ and $x_{i-1}<>1$ and $x_{i-1}<>n-1$ then return True;

if $x_t<>1$ then return True;

MR随机算法

- **Witness(a, n)**

```
{   x_0 ← ModularExponentiation(a,u,n);
    for i ← 1 to t do
    {   x_i ← (x_{i-1}^2 mod n);
        if x_i=1 and x_{i-1}<>1 and x_{i-1}<>n-1 then return True;
    }
    if x_t<>1 then return True;
    return False;
}
```

注: - 如果该算法返回True, 分二种情形:

1)从for循环里返回, 即找到了witness

2)从for循环外返回, 即n不满足Fermat定理

- 如果返回False, 未找到witness

MR随机算法

- **MillerRabin(n, s)**

```
{
    for j ← 1 to s do
    {   a ← random(1, n-1);
        if Witness(a, n) then
            return Compositive;
    }
    return Prime;
}
```

注: - 返回合数，结论是100%正确；

- 返回素数，存在误判的可能；
- 问题的是算法的复杂性和误判率。

计算示例

- 示例： **n=561是素数吗？**

  $n-1=560=2^4\times35$，取$a=7$作为测试基

  $x_0\equiv a^{35}\equiv 241 \pmod{561}$

  …

  $\Rightarrow X=\langle 241, 298, 166, 67, 1\rangle$

  最后一步表明，发现另外一个1的非平凡平方根，即

  $a^{280}\equiv 67 \pmod{561}$，$a^{560}\equiv 1 \pmod{561}$

  因此，$a=7$是n为合数的一个证人(Witness)

算法的复杂性和误判率

- 算法的复杂性

  设n为β-bit整数,

  算法需要$O(s\beta)$算术运算、 $O(\beta^3)$位运算。

- 算法的误判率

  - Th31.38

    n为奇合数， 则n的Witness数大于(n-1)/2。

  - Th31.39

    MR算法的误判率不超过2^{-s}。

MR v.s. Pseudo Prime

- 后者的误判率与n相关，不可控；
  前者仅与s相关，可控制误差；

- MR算法的优点不存在坏的输入，缺点取决于s和基a的选择。即witness带有幸运性质。

- MR算法是第1个多项式时间的随机算法，确定性算法直到2002年由三位印度科学家提出。

End of Chap31