内容回顾

- 计算模指数a^e **mod** m: 分治法
  - 二进制分拆e为$\underline{d_{n-1}d_{n-2}\ldots d_1d_0}$
  - 分治计算a^{D_i} **mod** m, $D_i = d_i * 2^i$
  - 归并
- 测试n是否为素数: 基于Eratosthenes筛选法
  - 确定待筛选集合$\{2, 3, \ldots, n^{1/2}\}$
  - 筛选$\{2, 3, \ldots, n^{1/2}\}$中所有素数
  - 判断$\{2, 3, \ldots, n^{1/2}\}$中的素数是否整除n

提示: modExp

```
int modExp(int a, int e, int m) {
  r = 1;
  while e is NOT 0
    if e % 2 = 1
      r = r * a mod m;
    end if
    a = a * a mod m;
    e = e >> 1;
  end while
  return r;
}
```

提示: Eratosthenes

- 确定待筛选集合set[0,...,m], m = $a^{1/2}$-1
  - ➡ **如果i在set中, 则set[i] = 1, 否则set[i] = 0**
- Eratosthenes筛选

```
for i from 2 to sqrt(a)
  j = i * i;
  while j ≤ m
    set[j] = 0;
    j = j + i;
  end while
end for
```

- 筛选后元素关于n的整除性判断

数论基础实验-素性检测

基于Eratosthenes筛选法的素性测试方法

Miller-Rabin素性测试方法

素数的两个性质

- 任意素数n都可表示为**n = 2^kq + 1**, k >= 0, q为奇数
  - 特殊情况: n = 2时, 2 = $2^0 * 1 + 1$
- n是素数, a是小于n的正整数, 则a^2 **mod** n = 1当且仅当 a **mod** n = 1或a **mod** n = n - 1
  - 充分性: a^2 **mod** n = (a **mod** n)(a **mod** n) **mod** n = 1
  - 必要性: a^2 **mod** n = 1, 则a^2-1 **mod** n = 0

    即(a+1)(a-1) **mod** n = 0

    由于**n为素数**, 因此a+1 **mod** n = 0或a-1 **mod** n = 0

    即a **mod** n = n-1或a **mod** n = 1

Miller-Rabin算法原理

❖ $n = 2^k q + 1$ ($k > 0$, q为奇数) 是大于2的素数, a是大于1且小于n-1的整数, 如下两个条件之一成立：

➡ a^q **mod** $n = 1$ 序列中所有项均为1

➡ 存在j ($j >= 1$且$j <= k$), 满足$a^{2^{j-1}q}$ **mod** $n = n-1$ 序列中存在一项为n-1, 使之后所有项均为1

★ 费马小定理: a^{n-1} **mod** $n = a^{2^k q}$ **mod** $n = 1$ =1

★ 序列: a^q **mod n**, a^{2q} **mod n**,..., $a^{2^{k-1}q}$ **mod n**, $a^{2^k q}$ **mod n**

✳ 后一项恰为前一项的平方: $a^{2^i q}$ **mod** $n = [(a^{2^{i-1}q}$ **mod** $n)^2]$ **mod** n

得证

Miller-Rabin素性测试算法

- **Miller-Rabin**(n)
  - ➡ 确定整数k和q, 满足$n = 2^k q + 1$
  - ➡ 随机选择整数a, 满足$a > 1$且$a < n-1$
  - ➡ **如果a^q mod n = 1, 返回“不确定” (可能是素数)**
  - ➡ **如果存在$a^{2^{j-1}q}$ mod n = n-1 (j = 1, 2,..., k), 返回“不确定”**
  - ➡ 返回“合数”
- 通过Miller-Rabin素性测试的数**不一定**是素数; 无法通过Miller-Rabin素性测试的数**一定不是**素数 (**必要条件**)

Miller-Rabin素性测试示例 I

判断29是否是素数

- 确定k和q: $29 = 2^2 * 7 + 1$, 因此, **$k = 2, q = 7$**
- 随机选择a: **$a = 2$**
  - ➡ **序列: 2^7 mod n, 2^{2*7} mod n, 2^{4*7} mod n**
  - ➡ **判定I:** a^q **mod** n = 12, 既不等于n-1, 又不等于1
  - ➡ **判定II:** 序列第二项a^{2q} **mod** n = 28 **= n-1**, 返回“不确定”

Miller-Rabin素性测试示例 II

判断221是否是素数

- 确定k和q: $221 = 2^2 * 55 + 1$, 因此, **k = 2, q = 55**
- 随机选择a: **a = 5**
  - ➡ 判定I: a^q **mod** n = 5^{55} mod 221 = 112, 既不等于n-1, 又不等于1
  - ➡ 判定II: 5^{2q} **mod** n = $(5^{55})^2$ mod 221 = 168, 返回**"合数"**

问题: 随机选择a = 21?
21^{55} mod 221 = 200
$(21^{55})^2$ mod 221 = 220, **"不确定?"**

非确定性测试
如果n不是素数, 选择不同的a, 测试结果不完全相同 **(多次测试)**

数论基础实验-乘法逆元

乘法逆元

- 定义: 对于整数a和m, 如果存在整数b, 满足**$a * b \bmod m = 1$**, 则称b为a关于模m的乘法逆元, 记为a^{-1}
- 乘法逆元目标: **已知a和m, 求解a^{-1}**
- 用途: 现代密码学加解密常涉及求解乘法逆元

乘法逆元存在的条件

- a存在关于模m的乘法逆元的充要条件是**a和m的最大公约数为1** (或a和m互素), 记为**gcd(a, m) = 1**

★ a与m互素, 则存在整数k_1, k_2, 满足$\mathbf{k_1*a + k_2*m = 1}$

- 等式两边同取**mod** m: $k_1*a = 1$ **mod** m, 即k_1是a关于模m的乘法逆元

乘法逆元存在条件示例

分析6和5关于模8的乘法逆元

- ➡ gcd(6, 8) = 2, gcd(5, 8) = 1

Z_8	0	1	2	3	4	5	6	7
乘以6	0	6	4	2	0	6	4	2
乘以5	0	5	2	7	4	1	6	3

6不存在关于模8的乘法逆元

5关于模8的乘法逆元为5(本身):

$5*5 + (-3)*8 = 1$

问题: 如何求解乘法逆元
$k_1*a + k_2*m = 1$, 求解k_1

欧几里德算法原理

- 最大公约数: gcd(a, m)是a和m的因子, 并且**a和m的任意因子都是gcd(a, m)的因子**
- 欧几里德算法: **辗转相除**, 求最大公约数

$a = q_1 * m + r_1$, $r_1 >= 0$且$r_1 < m$

如果$r_1 = 0$, gcd(a, m) = m; 否则, **gcd(a, m) = gcd(m, r_1)**

$m = q_2 * r_1 + r_2$, $r_2 >= 0$且$r_2 < r_1$, 意味着**gcd(m, r_1) = gcd(r_1, r_2)**

……辗转相除

$r_{n-1} = q_{n+1} * r_n + 0$, 那么**gcd(a, m) = gcd(m, r_1) = gcd(r_1, r_2) =…= r_n**

欧几里德算法示例 I

求7与96的最大公约数

- 假设$r_{-1} = 7$, $r_0 = 96$

i	r_i	q_i	公式
-1	7	0	$7 = 0 * 96 + 7$
0	96	13	$96 = 13 * 7 + 5$
1	7	1	$7 = 1 * 5 + 2$
2	5	2	$5 = 2 * 2 + 1$
3	2	2	$2 = 2 * 1$
4	1		

$r_4 = 1$, 因此$gcd(7, 96) = 1$

欧几里德算法示例 II

求270与96的最大公约数

- 假设r_{-1} = 270, r_0 = 96

i	r_i	q_i	公式
-1	270	2	$270 = 2 * 96 + 78$
0	96	1	$96 = 1 * 78 + 18$
1	78	4	$78 = 4 * 18 + 6$
2	18	3	$18 = 3 * 6$

gcd(270, 96) = 6

扩展欧几里德算法

❖ 在欧几里德算法的基础上，计算辗转相除的**系数**

$a = q_1 * m + r_1$		$\mathbf{1} * a + \mathbf{(-q_1)} * m = r_1$
$m = q_2 * r_1 + r_2$	$m + (-q_2) * r_1 = r_2$	$\mathbf{(-q_2)} * a + \mathbf{(1+q_1q_2)} * m = r_2$
$r_1 = q_3 * r_2 + r_3$	$r_1 + (-q_3) * r_2 = r_3$	$\mathbf{(1+q_2q_3)} * a + \mathbf{(1-q_1-q_1q_2q_3)} * m = r_3$
…	…	…
$r_{n-1} = q_{n+1} * r_n$	…	$\mathbf{k_1} * a + \mathbf{k_2} * m = r_n = 1$

欧几里得算法辗转相除

计算系数k_1, k_2, k_1为a关于模m的乘法逆元

$\boxed{r_{i-1} + (-q_{i+1}) * r_i = r_{i+1}}$

欧几里德扩展算法示例 I

求7关于模96的逆元: 假设$r_{-1} = 7$, $r_0 = 96$

i	r_i	q_i	公式		
-1	7	0	$7 = 0 * 96 + 7$		$\mathbf{1} * 7 + \mathbf{0} * 96 = 7$
0	96	13	$96 = 13 * 7 + 5$	$1 * 96 + (-13) * 7 = 5$	$\mathbf{(-13)} * 7 + \mathbf{1} * 96 = 5$
1	7	1	$7 = 1 * 5 + 2$	$7 + (-1) * 5 = 2$	$\mathbf{(14)} * 7 + \mathbf{(-1)} * 96 = 2$
2	5	2	$5 = 2 * 2 + 1$	$5 + (-2) * 2 = 1$	$\mathbf{(-41)} * 7 + \mathbf{3} * 96 = 1$
3	2	2	$2 = 2 * 1$		
4	1				

$\mathbf{k_1 = -41, k_2 = 3}$

可选: 将k_1变换为Z_{96}中的元素55

欧几里德扩展算法示例 II

求270关于模96的乘法逆元: 假设r_{-1} = 7, r_0 = 96

i	**r_i**	**q_i**	**公式**		
-1	270	0	270 = 2 * 96 + 78		**1***270 + **(-2)***96 = 78
0	96	1	96 = 1 * 78 + 18	1*96 + (-1)*78 = 18	**(-1)***270 + **3***96 = 18
1	78	4	78 = 4 * 18 + 6	78 + (-4)*18 = 6	**5***270 + **(-14)***96 = 6
2	18	3	18 = 3 * 6	18 + (-3)*6 = 0	

由于gcd(270, 96) != 1, 不存在乘法逆元

k_1 = 5, k_2 = -14

扩展阅读

- **Prime Numbers, Factorization and Euler Function**
- **Primality Testing : Non-deterministic Algorithms**（费马小定理、Miller-Rabin测试、Solovay-Strassen测试）
  - ➡ 利用素数的一般性质实施的**必要性**测试
- 欧几里德算法的三种实现思路
- 欧几里德算法的应用: **一道ACM试题; 解答**

课堂实验

```
bool millerRabin(int a)
// a: 输入测试数
// 如果a是合数，返回0；否则返回1
```

```
int euclid(int a, int m)
// a: 输入正整数; m: 输入模数(m>a)
// 返回a关于m的乘法逆元
```