中国计算机学会
China Computer Federation

初等数论

绍兴市第一中学　　翁天东

初等数论是研究数的规律，特别是整数性质的数学分支。它是数论的一个最古老的分支。它以算术方法为主要研究方法，主要内容有整数的整除理论、同余理论、不定方程、数论函数等。

质数-定义及性质

质数又称素数，是指在大于1的自然数中，除了1和它本身以外不能被其他自然数整除的数；否则称为合数。

规定1既不是质数也不是合数。

质数的个数有无穷个。(欧几里得《几何原本》)

素数定理(the Prime Number Theorem,PNT)：设$x \geq 1$，以$\pi(x)$表示不超过x的素数的个数。

当$x \to \infty$时，$\pi(x) \to \frac{x}{\ln(x)}$

质数判定-试除法

判断n是否是质数：根据定义，枚举 $x \in [2, n-1]$，判断是否存在 $x|n$ 。

考虑到不存在$x \in \left[\left\lfloor\frac{n}{2}\right\rfloor + 1, n-1\right], x|n$，因此$x \in [2, \left\lfloor\frac{n}{2}\right\rfloor]$。

考虑若$k|n$，则存在n的另一个因子$\frac{n}{k}$满足$k \le \sqrt{n} \le \frac{n}{k}$，因此枚举$x \in [2, \sqrt{n}]$即可。

```cpp
bool is_prime(int n) {
        for (int i=2;i*i<=n;i++)
                if (n%i==0) return 1;
        return 0;
}
```

质数筛选-埃式筛法

暴力筛法：枚举$x \in [2, n]$，用is_prime()判断x是否是质数。

埃式筛法：显然，对任意整数x，其倍数2x,3x,…都是合数；一个合数一定能被分解为若干质数的积。

从小到大依次枚举质数，将$[1, n]$内的所有倍数都标记为合数；一个数如果没有被标记为合数就是质数。

```
for (int i=2;i<=n;i++)
        if (!f[i]){
                prime[++cnt]=i;
                for (int j=i+i;j<=n;j+=i) f[j]=1;
        }
```

质数筛选-欧拉筛法

埃氏筛中，一个合数会被它的所有质因子重复标记，理想的筛法应当是一个合数只会被标记一次。

欧拉筛法(线性筛)：保证对任一合数，只会被其最小质因数标记。

对于每一个数i，从小到大枚举当前质数集 $p, p_j \le i$，标记合数 $i \times p_j$，若$p_j|i$说明$i = p_j \times u$；对于$p_j < p_k \le i$，$i \times p_k = p_j \times u \times p_k$，当$i = u \times p_k$时能被$p_j$筛掉的，筛质数的过程就不需要$p_k$参与了。

```
for (int i=2;i<=n;i++) {
        if (!f[i]) prime[++cnt]=i;
        for (int j=1;j<=cnt&&i*prime[j]<=n;j++) {
                f[i*prime[j]]=0;
                if (i%prime[j]==0) break;
        }
}
```

举个例子：i=8，p_1=2，此时已存在2|8，若继续枚举素数集p_2=3，标记8×3=24=2×12，便可发现由于24含有的因子8含有质因子2，所以必然会在i=12时由最小质数2去标记，不需要质数3在i=8时去掺一手。

质因数分解

唯一分解定理(算术基本定理)：任何一个大于1的自然数N一定可以分解为一个或几个质数的积：

$$N = {p_1}^{r_1} \cdot {p_2}^{r_2} \cdot \cdots \cdot {p_n}^{r_n}$$

算法实现：从小到大依次枚举N的因数i，若$i|N$，将N除以i至除不尽，若$i^2 > N$退出循环。

① i显然只能是质数，如果是合数前面已经被除尽了；

② 退出循环后N>1则剩下的N也是一个质数

```
for (int i=2;i*i<=N;i++)
 if (N%i==0) {
  prime[++cnt]=i,d[cnt]=0;
  while (N%i==0) N/=i,d[cnt]++;
 }
if (N>1) prime[++cnt]=N,d[cnt]=1;
```

GCDS-Sabbir and gcd Problem(SPOJ30919)

Sabbir is a little boy. He loves math very much. One day his friend Taskin gave him a very hard task. Taskin gave him **n** numbers $\mathbf{a_1, a_2, a_3, \ldots a_n}$

Taskin asked for a minimum integer number **x** (**x** > **1**) such that $\mathbf{gcd(x, a_1) = 1, gcd(x, a_2) = 1, \ldots gcd(x, a_n) = 1}$.

In other words you have to find a minimum integer **x** (**x** > **1**) such that

$$\forall i,\ i \in [1...n],\ gcd(x, a_i) = 1$$

Note: gcd is greatest common divisor.

机翻：

给定n个整数a_i，找到最小的整数x使得他与所有的a_i都互质且x>1。

多组数据，1<=t<=10；1<=n<=10^5，1<=a_i<=10^7。

例题

给定n个整数a_i，找到最小的整数x使得他与所有的a_i都互质且x>1。
多组数据，1<=t<=10；1<=n<=10^5，1<=a_i<=10^7。

算法：

所求x要取最小值，且与所有a_i互质，质数的答案更优秀。

如果存在符合条件的合数解，对合数分解质因数必然能分解出与a_i互质的质因数，这个质因数比合数解更小。

需要找一个和所有a_i互质的最小质数，显然要先做一遍欧拉筛。

对所有a_i进行质因数分解，没被标记过的最小质数就是答案。

考虑减少质因数分解复杂度，减少无效枚举，可以在欧拉筛时，维护数组pre[a*b]=b，其中$a=p^r$，p是a*b最小的质因数。

```
if (i%prime[j]==0) { pre[i*prime[j]]=pre[i]; break; }
                    else pre[i*prime[j]]=i;
```

对一个a_i的分解复杂度与其质因数个数相关，最多8次就分解完了。

约数

约数，也称为因数。整数a除以整数b求得的商正好是整数且没有余数，a能被b整除，或者说b能整除a，写作$b|a$。a是b的倍数，b是a的约数。

$$N = {p_1}^{r_1} \cdot {p_2}^{r_2} \cdot \dots \cdot {p_n}^{r_n}$$

根据唯一分解定理，一个整数N的约数m可表示为：

$$\mathrm{m} = {p_1}^{d_1} \cdot {p_2}^{d_2} \cdot \dots \cdot {p_n}^{d_n}, 0 \le d_i \le r_i$$

可得N的约数个数公式：

$$(r_1 + 1) \times (r_2 + 1) \times \dots \times (r_n + 1) = \prod_{i=1}^{n} (r_i + 1)$$

约数获取的程序实现：参照因数分解的思路即可

公约数、公倍数

公约数：若整数a同时是整数b、c的约数，则称a为b与c的公约数(Common Divisor)。

a,b最大的公约数(Greatest Common Divisor)写作gcd(a,b)。

特别的，$\gcd(a,0)=a$；$\gcd(a,b)=1$称a,b互质。

最小公倍数(Least Common Multiple)：整数a与b最小的公倍数，写作lcm(a,b)

$$\operatorname{lcm}(a,b)=a\times b\div\gcd(a,b)$$

求最大公约数算法：更相减损术=>辗转相除法

辗转相除法(欧几里得算法)：gcd(a,b)=gcd(b,a mod b)

```
int gcd(int a,int b) { return !b?a:gcd(b,a%b); }
```

欧拉函数

欧拉函数φ(n)($n \in N^*$)表示小于等于n的正整数中与n互质的数的个数。
给出数学定义如下：

$$\varphi(\mathrm{n}) = \sum_{i=1}^{n}[\gcd(i,n) = 1]$$

若对n分解质因数使得$\mathrm{n} = {p_1}^{k_1}{p_2}^{k_2}\cdots{p_m}^{k_m}$，可得：

$$\frac{\varphi(n)}{n} = \prod_{i=1}^{m}(1-\frac{1}{p_i})$$

如求1000的欧拉函数φ(1000)可得：

$$\varphi(1000) = \varphi(2^3 \times 5^3) = 1000\left(1-\frac{1}{2}\right)\left(1-\frac{1}{5}\right) = 400$$

即[1,1000]中与1000互质的数有400个。

欧拉函数-证明1

设p_1,p_2是n质因数，则在[1,n]有n/p_1个p_1倍数，有n/p_2个p_2倍数。

根据容斥原理，可得[1,n]中不能被p_1,p_2整除的数共有：

$$n-\left(\frac{n}{p_1}+\frac{n}{p_2}-\frac{n}{p_1 \times p_2}\right)=n\times(1-\frac{1}{p_1})\times(1-\frac{1}{p_2})\text{个}$$

归纳可得，即对于n的所有质因数$p_1,p_2,\cdots,p_m$可得：

$$\frac{\varphi(n)}{n}=\prod_{i=1}^{m}(1-\frac{1}{p_i})$$

欧拉函数-证明2

若n=1，则$\varphi(1) = 1$；

若n是质数，则$\varphi(n) = n - 1$，因为质数与小于它的每个正整数都互质；

若$n = p^k$（p为质数，$k \in N^*$），则小于等于n的数中，因子包含质数p(也就是不存在互质)的数共计p^{k-1}个，即$p \times 1, p \times 2, \cdots, p \times p^{k-1}$，剩余与$p^k$互质的数为：

$$\varphi(p^k) = p^k - p^{k-1} = p^k(1 - \frac{1}{p})$$

若$n = p \cdot q$，而且p，q互质，有$\varphi(n) = \varphi(p \cdot q) = \varphi(p) \cdot \varphi(q)$ （欧拉函数是积性函数）[积性函数指对于所有互质的整数a和b有性质$f(a \cdot b) = f(a) \cdot f(b)$的数论函数]

中国计算机学会 China Computer Federation

欧拉函数-证明2

证明： 将[1,p×q]的值排列如下

$$
\begin{matrix}
1 & 2 & \cdots & i & \cdots & q \\
1+q & 2+q & \cdots & i+q & \cdots & 2q \\
1+2q & 2+2q & \cdots & i+2q & \cdots & 3q \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
1+jq & 2+jq & \cdots & i+jq & \cdots & (j+1)q \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
1+(p-1)q & 2+(p-1)q & \cdots & i+(p-1)q & \cdots & pq
\end{matrix}
$$

对于每一行数$\mathrm{i+jq}$，其对q取余的余数为[1,2,3,⋯,q−1,0]，即有$\varphi(q)$个数与q互质；对于每一列数$\mathrm{i+jq, j \in [0,p-1]}$，由于p、q互质，同理也有$\varphi(\mathrm{p})$个数与p互质。

对于任一与$p \times q$互质的数a($a < p \times q$)，若a与p互质，a与q互质，则a属于这$\varphi(\mathrm{q})$行、$\varphi(\mathrm{p})$列中的某一数。这样的a总共有$\varphi(\mathrm{p}) \cdot \varphi(\mathrm{q})$个。即

$$\varphi(\mathrm{p} \cdot \mathrm{q}) = \varphi(\mathrm{p}) \cdot \varphi(\mathrm{q})$$

欧拉函数-证明2

通式： 对于任意一个非1的正整数，都可写成一系列质数之积：

$$n = p_1^{k_1} \cdot p_2^{k_2} \cdots p_m^{k_m}$$（$p_1, \cdots p_m$都为质数）

由欧拉函数的积性性质$\varphi(p \cdot q) = \varphi(p) \cdot \varphi(q)$可得

$$\varphi(n) = \varphi(p_1^{k_1})\varphi(p_2^{k_2}) \cdots \varphi(p_m^{k_m})$$

由式$\varphi(p^k) = p^k - p^{k-1} = p^k(1-\frac{1}{p})$可得

$$\varphi(n) = p_1^{k_1}(1-\frac{1}{p_1})p_2^{k_2}(1-\frac{1}{p_2}) \cdots p_m^{k_m}(1-\frac{1}{p_m})$$

$$\varphi(n) = p_1^{k_1}p_2^{k_2} \cdots p_m^{k_m} \cdot \left(1-\frac{1}{p_1}\right)\left(1-\frac{1}{p_2}\right)\cdots\left(1-\frac{1}{p_m}\right)$$

即$\varphi(n) = n \cdot \prod_{i=1}^{m}(1-\frac{1}{p_i})$

欧拉函数-性质

- $\varphi(a \cdot b) = \varphi(a) \cdot \varphi(b) \cdot \frac{d}{\varphi(d)}, d = \gcd(a, b)$
- $[1, n]$中所有与n互质的数之和为$\frac{\varphi(n) \times n}{2}$
- $\sum_{i|n} \varphi(i) = n$
- $\varphi(n) = \begin{cases} \varphi(\frac{n}{p}) \times p, & p|n, p^2|n \\ \varphi(\frac{n}{p}) \times (p-1), & p|n, p^2 \nmid n \end{cases}$

欧拉函数-算法实现

正常求φ(n)就是在分解质因数时求解：

```
inline int Phi(int n)
{   int i,ans=n;
    for (i=2;i*i<=n;i++)
    if (n%i==0)
    {   ans=ans/i*(i-1);
        while (n%i==0) n/=i;
    }
    if (n>1) ans=ans/n*(n-1);
    return ans;
}
```

参考欧拉筛可以在O(n)求出[1,n]的φ函数的值：

```
inline void Phi(int n)
{   int i,j;
    for (phi[1]=1,i=2;i<=n;i++)
    {   if (!f[i]) phi[i]=i-1,b[++l]=i;
        for (j=1;j<=l;j++)
        {   if (b[j]>n/i) break;
            f[b[j]*i]=1;
            if (i%b[j]==0) { phi[b[j]*i]=phi[i]*b[j]; break; }
            phi[b[j]*i]=phi[i]*(b[j]-1);
        }
    }
}
```

中国计算机学会 China Computer Federation

例题

Longge's Problem(Poj2480)

Longge is good at mathematics and he likes to think about hard mathematical problems which will be solved by some graceful algorithms. Now a problem comes: Given an integer N(1<N<2^31),you are to calculate ∑gcd(i, N) 1<=i <=N.

"Oh, I know, I know!" Longge shouts! But do you know? Please solve it.

机翻：

龙哥擅长数学，他喜欢思考一些困难的数学问题，这些问题可以通过一些优美的算法来解决。现在问题来了：给定一个整数N(1<N<2^{31})，你需要计算∑gcd(i,N)，1<=i<=N。

例题

算法1：

$$\sum_{i=1}^{n}\gcd(i,n)$$
$$=\sum_{d|n}d\cdot\sum_{i=1}^{n}[\gcd(i,n)==d]$$
$$=\sum_{d|n}d\cdot\sum_{j=1}^{\frac{n}{d}}[\gcd(j\times d,n)==d]$$
$$=\sum_{d|n}d\cdot\sum_{j=1}^{\frac{n}{d}}[\gcd\left(j,\frac{n}{d}\right)==1]$$
$$=\sum_{d|n}d\cdot\varphi(\frac{n}{d})$$

$O(\sqrt{d})$枚举n的约数d，求出相应的$\varphi(\frac{n}{d})$即可。

算法2：

在n确定的情况下，容易想到，若i、j互质，gcd(i,n)、gcd(j,n)也互质，即gcd(i×j,n)=gcd(i,n)×gcd(j,n)，说明gcd()是一个积性函数。

而$F(n)=\sum_{i=1}^{n}\gcd(i,n)$满足其积性，证明略。

根据F(n)的积性列出n的唯一分解式：

$$\mathrm{F(n)}=\mathrm{F}\left(p_1{}^{k_1}\right)\mathrm{F}\left(p_2{}^{k_2}\right)\cdots\mathrm{F}\left(p_m{}^{k_m}\right)$$

对于单个质因子的指数p^r来说，其因子分别为$1,p^2,\cdots,p^r$，最大因子和可计算：

$$=1\times(p^r-p^{r-1})+p\times(p^{r-1}-p^{r-2})+\cdots+p^r\times 1$$
$$=(r+1)\times p^r-r\times p^{r-1}$$

对n分解质因数，将每一组的$\mathrm{F}(p_i{}^{k_i})$相乘即可。

不定方程

所谓不定方程，是指未知数的个数多于方程个数，且未知数受到某种限制（如要求是有理数、整数或正整数等等）的方程或方程组。

二元一次不定方程：就是形如 $ax+by=c$ 的方程，其中a，b，c已知。

解法：

1、判断是否有解(裴蜀定理：a、b是不全为0的整数，则存在整数x,y，使得$ax+by=\gcd(a,b)$)

若$c \bmod \gcd(a,b) \neq 0$，那么方程不存在整数解。

2、化简、转化

方程可转化为$a'x+b'y=c'$，

其中$a'=\frac{a}{\gcd(a,b)}$，$b'=\frac{b}{\gcd(a,b)}$，$c'=\frac{c}{\gcd(a,b)}$。

3、求一组特解

此时需用到Exgcd

扩展欧几里得

扩展欧几里得(exgcd)定理：

对于两个不全为0的整数a、b，必存在一组解x,y，使得$ax+by=\gcd(a,b)$。

对于方程$a'x+b'y=c'$，我们知道gcd有一个性质为$\gcd(a,b)=\gcd(b,a \bmod b)$，若辗转相除法求到最后，$b'$将等于0，此时$x=\frac{c'}{a'}$，y=0。这就求出了一组特解。

而对于方程$ax+by=\gcd(a,b)$，可借exgcd代码实现：

```
int exgcd(int a,int b)
{ if (!b) { x=1,y=0; return a; }
 int d=exgcd(b,a%b),t=x;
 x=y;
 y=t-(a/b)*y;
 return d;
}
```

扩展欧几里得-代码解析

```
int exgcd(int a,int b)
{ if (!b) { x=1,y=0; return a; }
 int d=exgcd(b,a%b),t=x;
 x=y;
 y=t-(a/b)*y;
 return d;
}
```

由扩展欧几里得定理：$ax + by = \gcd(a, b)$

可知若此时b=0，说明$\gcd(a, 0) = a$。

原式变为$ax + by = a \to x = 1$，$y = 0$。

扩展欧几里得-代码解析

#define % mod

```
int exgcd(int a,int b)
{ if (!b) { x=1,y=0; return a; }
 int d=exgcd(b,a%b),t=x;
 x=y;
 y=t-(a/b)*y;
 return d;
}
```

设x,y表示第一次递归时的值，x_1,y_1表示第二次递归时的值。$\mathrm{gcd}(a,b)=\mathrm{gcd}(b,a\%b)$代入exgcd定理，有$ax+by=b\times x1+(a\%b)\times y1$。将右式变形：

$$
\begin{aligned}
& b\times x_1+(a\%b)\times y_1 \\
&= b\times x_1+\left(a-b\times\left\lfloor\frac{a}{b}\right\rfloor\right)\times y_1 \\
&= a\times y_1+b\times\left(x_1-\left\lfloor\frac{a}{b}\right\rfloor\times y_1\right)
\end{aligned}
$$

最终得到$ax+by=a\times y_1+b\times\left(x_1-\left\lfloor\frac{a}{b}\right\rfloor\times y_1\right)$。

也就是说，上一深度的x等于下一深度的y_1，上一深度的y等于下一深度的$x_1-\left\lfloor\frac{a}{b}\right\rfloor\times y_1$。

扩展欧几里得-特解与通解

到这里为止，我们便得到了不定式 $ax + by = \gcd(a, b)$ 的一组特解x、y。
对于一般的不定式$ax + by = c$，解应该是：

$$x_0 = x \times \left(\frac{c}{\gcd(a,b)}\right), \quad y_0 = y \times \left(\frac{c}{\gcd(a,b)}\right)$$

构造通解：

对于需要求出所有整数解的情况，设$d = \gcd(a, b)$。则

$$x = x_0 + \frac{b}{d} * t, \quad y = y_0 - \frac{a}{d} * t, \quad t \in Z$$

同余方程(NOIP2012T4)

现有整数a,b满足$ax\%b = 1$，要求求出最小的正整数x;
分析：式子可以转化成$ax + by = 1$然后求exgcd后x最小值。

$$ax \equiv 1 \pmod b$$

同余方程是一个数学方程式。该方程式的内容为：对于一组整数Z，Z里的每一个数都除以同一个数p，得到的余数可以为0,1,2,3……p-1共p种。我们就以余数的大小作为标准将Z分为p类。每一类都有相同的余数。

单个同余方程解：

求$ax \equiv b(\mathrm{mod}\ p)$关于未知数x的解。
该式可转换成$ax + py = b$，$y \in Z$，即可套用exgcd模板求解。

同余-性质

取模运算规则：

$$(a+b)\ \%\ p = (a\ \%\ p + b\ \%\ p)\ \%\ p$$

$$(a-b)\ \%\ p = (a\ \%\ p - b\ \%\ p)\ \%\ p$$

$$(a\times b)\ \%\ p = (a\ \%\ p \times b\ \%\ p)\ \%\ p$$

$$(a^b)\ \%\ p = \left((a\ \%\ p)^b\right)\ \%\ p$$

若$a \equiv b \pmod{p}$，那么$a^n \equiv b^n \pmod{p}$

若 $a \bmod p = x$，$a \bmod q = x$，且 p、q 互质，则 $a \bmod (p\times q) = x$

结合律：

$$\left((a+b)\ \%\ p + c\right)\ \%\ p = (a + (b+c)\ \%\ p)\ \%\ p$$

$$\left((a\times b)\ \%\ p \times c\right)\ \%\ p = (a \times (b\times c)\ \%\ p)\ \%\ p$$

交换律：

$$(a+b)\ \%\ p = (b+a)\ \%\ p$$

$$(a\times b)\ \%\ p = (b\times a)\ \%\ p$$

分配率：

$$\left((a+b)\ \%\ p \times c\right)\ \%\ p = \left((a\times c)\ \%\ p + (b\times c)\ \%\ p\right)\ \%\ p$$

欧拉定理

在数论中，欧拉定理（也称费马-欧拉定理）是一个关于同余的性质。欧拉定理表明，若n,a为任意互质的正整数，则：

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

欧拉定理-证明1

对于集合$Z_n=\{x_1,x_2,\cdots,x_{\varphi(n)}\}$，其中$x_i(i=1,2,\cdots,\varphi(n))$是不大于n且与n互质的数。

考虑构造集合$S=\{ax_1 \pmod n, ax_2 \pmod n,\cdots,ax_{\varphi(n)} \pmod n\}$

① $\forall i\in[1,\varphi(n)]$，由于$a,n$互质，$x_i$也与$n$互质，则$a\times x_i \pmod n$也一定与$n$互质。

因此$\forall x_i$，$a\times x_i \pmod n$必然也是Z_n的其中一个元素。所以S中所有元素都与n互质，都小于n，且都相当于Z_n的其中一个元素。

② 对于Z_n中两个元素x_i和x_j，如果$x_i\neq x_j$

则$a\times x_i \pmod n\neq a\times x_j \pmod n$，这个可由a、n互质得出。

即S是存在$\varphi(n)$个互不相同且与n互质的可以与集合Z_n元素一一对应的集合。

所以，很明显$S=Z_n$

欧拉定理-证明2

即S是存在φ(n)个互不相同且与n互质的可以与集合Z_n元素一一对应的集合。

所以，很明显$S=Z_n$

既然这样，可得：

$$\prod_{i=1}^{\varphi(n)} ax_i \equiv \prod_{i=1}^{\varphi(n)} x_i \ (\,mod\; n)$$

$$a^{\varphi(n)} \prod_{i=1}^{\varphi(n)} x_i \equiv \prod_{i=1}^{\varphi(n)} x_i \ (\mathrm{mod}\; \mathrm{n})$$

由于$x_1, x_2, \cdots, x_{\varphi(n)}$都与$n$互质，$\prod_{i=1}^{\varphi(n)} x_i$必然也与$n$互质，所以等式两边抵消为：

$$a^{\varphi(n)} \equiv 1 \ (\mathrm{mod}\; \mathrm{n})$$

欧拉定理-推论

推论1：

对于互质的数a,n，满足$a^{\varphi(n)+1} \equiv \mathrm{a} \ (\mathrm{mod}\ \mathrm{n})$

推论2：

对于互质的数a,n，满足$\mathrm{a}^{b} \equiv a^{b\ mod\ \varphi(n)}\ (mod\ n)$

设$\mathrm{b} = \mathrm{k} \times \varphi(\mathrm{n}) + \mathrm{r}$，则$\mathrm{r} = \mathrm{b} \bmod \varphi(\mathrm{n})$。

$\mathrm{a}^{b} \equiv a^{\mathrm{k} \times \varphi(\mathrm{n}) + \mathrm{r}} \equiv (a^{\varphi(n)})^{k} \times a^{r} \equiv 1^{k} \times a^{\mathrm{b}\ \mathrm{mod}\ \varphi(\mathrm{n})} \equiv a^{b\ mod\ \varphi(n)}\ \ (mod\ n)$

费马小定理

若a是与质数p互质的正整数，则有$a^{p-1} \equiv 1 \ (mod\ p)$。

费马小定理通常用来检验一个数是否是质数，是质数的必要非充分条件。也可用来进行快速幂的降幂操作。

当然满足费马小定理检验的数未必是质数，这种合数叫做卡迈克尔数。

证明1：

由于p是质数，所以有$\varphi(p) = p - 1$，代入欧拉定理即可证明。

费马小定理-证明2

该定理可改写成[若a是与质数p互质的正整数，则有$a^p \equiv a(mod\, p)$]
即$a^p - a \equiv 0(mod\, p)$。
如果p是质数，那么$p|a^p - a$对于任意整数a都成立。

用归纳法证明：假设$p|a^p - a$，那么

$(a+1)^p-(a+1) = \sum_{r=1}^{p-1}\binom{p}{r}a^r + a^p - a$，由$p|a^p - a$，所以

$(a+1)^p-(a+1) \equiv \sum_{r=1}^{p-1}\binom{p}{r}a^r \pmod p$

当$1 \le r \le p-1$时，二项式系数$\binom{p}{r} = \frac{p!}{(p-r)!r!}$的分子中有p，而分母中每一个因子都不能整除p（因为p是质数），所以$p|\binom{p}{r}$，因此

$(a+1)^p-(a+1) \equiv 0 \pmod p$

得到$p|(a+1)^p - (a+1)$。该式当$a \ne 0$成立，所以该假设成立。

逆元

满足$a \times k \equiv 1 \pmod p$的k值就是a关于p的乘法逆元inv[a]。（a、p要互质）

为什么要有乘法逆元呢？

当我们要求$a/b \bmod p$的值，且$b|a$，但由于a、b太大无法直接求得$a/b \pmod p$的值时，我们就要用到乘法逆元。

若存在$a \times k \equiv 1$ (mod p，a、p互质)，

则该式等同于$a \times a^{-1} \equiv 1$ ($mod\ p$，a、p互质)。此时a的逆元k等效于乘上a的倒数，即等效于除数a。

因此，我们可以通过求b关于p的乘法逆元k，将a乘上k再模p，即$(a \times k)\ mod\ p$。其结果与$(a/b)\ mod\ p$等价。

逆元-证明

证明1：

根据$b \times k \equiv 1 \ (\mathrm{mod}\ p)$有$b \times k = p \times x + 1$。

$k = (p \times x + 1)/b$。

把k代入$(a \times k)\ mod\ p$，得：

$$
\begin{aligned}
&\left(a \times \frac{p \times x + 1}{b}\right)\ mod\ p \\
&= \left(\frac{a \times p \times x}{b} + \frac{a}{b}\right) mod\ p \\
&= \left[\frac{a \times p \times x}{b} \mathrm{mod}\ p + \frac{a}{b}\right] mod\ p \\
&= \left[p \times \frac{a \times x}{b}\ \mathrm{mod}\ p + \frac{a}{b}\right] mod\ p
\end{aligned}
$$

因为$\left(p \times \frac{a \times x}{b}\right)\ mod\ p = 0$

所以原式等于：$(a/b)\ mod\ p$

证明2：

由逆元定义式$b \times \mathrm{inv}[b] \equiv 1 \ (\mathrm{mod}\ p)$

两边皆乘$\frac{a}{b}$，可得$\frac{a}{b} \times b \times inv[b] \equiv \frac{a}{b}\ (mod\ p)$

即$a \times \mathrm{inv}[b] \equiv a \div b \ (\mathrm{mod}\ p)$

逆元-算法实现

Exgcd求逆元：

因为$a \times k \equiv 1 \pmod p$，可转成$ax + py = 1$，$x,y \in Z$。

这个可用exgcd求出，原同余方程的唯一解就是用扩展欧几里德算法得出的x。

欧拉函数求逆元：

若a是与p互质的正整数，则有$a^{\varphi(p)} \equiv 1 \ (mod\ p)$。

则$a^{\varphi(p)-1} \equiv a^{-1} \ (mod\ p)$

所以$a^{\varphi(p)-1} \ (mod\ p)$即为a的逆元。

逆元-算法实现

费马小定理求逆元：

若a是与质数p互质的正整数，则有$a^{p-1} \equiv 1\ (mod\ p)$。

则$a \times a^{p-2} \equiv 1\ (mod\ p)$

所以$a^{p-2}\ (\mathrm{mod\ p})$即为a的逆元。

```
ll inv(int n)
{    if (n==1) return a;
     ll t=inv(n>>1)%p; t*=t,t%=p;
     if (n&1) return t*a%p;
     return t;
}
//gcd(a,p)=1&&p==prime[]
//a%=p,inv(p-2)
```

逆元-算法实现

逆元递推公式：

求解[1,p-1]模质数p的所有逆元，可以O(n)的递推算法实现。

对于任意小于p的数i，存在表达式$p = k \times i + r \rightarrow k \times i + r \equiv 0 \pmod p$。

其中$k = \left\lfloor \frac{p}{i} \right\rfloor$，$r = p \bmod i$。

调整表达式为$r \equiv -k \times i \pmod p$

两边同乘$r^{-1}i^{-1}$得$r \times inv[r] \times inv[i] \equiv -k \times i \times inv[i] \times inv[r] \pmod p$

$inv[i] \equiv -k \times inv[r] \pmod p$，由于$-k \equiv p - k \pmod p$

可得递推式$inv[i] = (p - k) \times inv[r] \bmod p = (p - p/i) \times inv[p \bmod i] \bmod p$

阶乘逆元递推：

因为$n! = (n-1)! \times n$，所以$\frac{1}{(n-1)!} = \frac{1}{n!} \times n$。

M斐波那契数列(Hdu4549)

M斐波那契数列F[n]是一种整数数列，它的定义如下：

F[0] = a
F[1] = b
F[n] = F[n-1] * F[n-2] (n > 1)

现在给出a, b, n【多组数据】 (0 <= a, b, n <= 10^9)， 你能求出F[n]的值吗？

例题

算法：

通过题目给出条件再往后推几项就可确定通项公式：

$$F[n]=a^{f[n-1]}\times b^{f[n]},n>0,f[n]=f[n-1]+f[n-2]$$

最终答案要取模，模数$p=10^9+7$，是一个质数。根据费马小定理，若p是质数，且a, b与p互质，则$F[n]\equiv a^{f[n-1]\%(p-1)}\times b^{f[n]\%(p-1)},n>0(mod\ p)$

总结：矩阵乘法求f[n]，用费马小定理减小指数后，再对该式求一遍快速幂即可。

同余方程组

对于同余方程组

$$\begin{cases} x \equiv b_1 (\mathit{mod}\ p_1) \\ x \equiv b_2 (\mathit{mod}\ p_2) \\ \quad \dots\dots \\ x \equiv \mathrm{b}_n (\mathrm{mod}\ \mathrm{p}_n) \end{cases}$$

$(p_1，p_2，\cdots，p_n$互质$)$

必定存在有解的条件，并可用构造法给出有解情况下解的具体形式。

中国剩余定理

设整数p_1，p_2，…，p_n两两互质，则对任意的整数b_1，b_2，…，b_n，方程组有解，并且通解可通过如下方式构造得到：

设$P = p_1 \times p_2 \times \cdots \times p_n = \prod_{i=1}^{n} p_i$是整数$p_1$，$p_2$，…，$p_n$的乘积，

并设$P_i = \frac{P}{p_i}$，$\forall i \in \{1,2,\cdots,n\}$是除了$p_i$以外的n-1个整数的乘积。

设$t_i = P_i^{-1}$为P_i模p_i的数论倒数。（t_i为P_i模p_i意义下的逆元）

$$P_i t_i \equiv 1 \ (mod\ p_i),\ \forall i \in \{1,2,\cdots,n\}$$

则方程组的通解形式为

$$x = b_1 t_1 P_1 + b_2 t_2 P_2 + \cdots + b_n t_n P_n + kP = kP + \sum_{i=1}^{n} b_i t_i P_i,\ k \in Z$$

在模P的意义下，方程组存在一个特解：$x = (\sum_{i=1}^{n} b_i t_i P_i)\ mod\ P$。

中国剩余定理-证明

由假设可知$\forall i \in \{1,2,\cdots,n\}$，$\forall j \in \{1,2,\cdots,n\}$，$j \neq i$， $\gcd(p_i, p_j) = 1$， $\gcd(p_i, P_i) = 1$。

存在整数t_i使得$t_i P_i \equiv 1 \ (mod\ p_i)$。可知乘积

$$b_i t_i P_i \equiv b_i \cdot 1 \equiv b_i \ (mod\ p_i),$$

$$\forall j \in \{1,2,\cdots,n\}，\ x = b_i t_i P_i + \sum_{j \neq i} b_j t_j P_j \equiv b_i + \sum_{j \neq i} 0 \equiv b_i \ (mod\ p_i)$$

可见x为方程组的一个解。

中国剩余定理-解

此外，设x_1和x_2都是方程组的解，那么：

$$\forall i \in \{1,2,\cdots,n\},\ x_1 - x_2 \equiv 0 \pmod{p_i}$$

由于p_1，p_2，⋯，p_n两两互质，说明$P = \prod_{i=1}^{n} p_i$整除$x_1 - x_2$，因此方程组的任意两解之间必然相差P的整数倍。所以方程组的解集是

$$\{kP + \sum_{i=1}^{n} b_i t_i P_i,\ k \in Z\}$$

对于模数不是质数，有时可用CRT求解，例：$a \bmod P = ?$

$$P = \prod_{i=1}^{n} p_i, \gcd(p_i, p_j) = 1$$

$$a \bmod \{p_1, \ldots, p_n\} = \{a_1, \ldots, a_n\}$$

就可以用CRT求了，见【SDOI2010】古代猪文。

扩展中国剩余定理

对于同余方程组

$$\begin{cases} x \equiv b_1(\,mod\; p_1) \\ x \equiv b_2(\,mod\; p_2) \end{cases}$$

(p_1，p_2，…，p_n并不两两互质)

由于中国剩余定理具有p_i必须两两互质的特殊限制，所以必须存在一个运用更广泛的算法来求出任意线性同余方程组的解。

扩展中国剩余定理-算法实现

先直接考虑两个方程：

$$\begin{cases} x - p_1 \times y_1 = b_1 \\ x - p_2 \times y_2 = b_2 \end{cases}$$

其中下式减上式得$p_1 \times y_1 - p_2 \times y_2 = b_2 - b_1$

再用exgcd可求出y_1和y_2，此时$x' = b_1 + p_1 \times y_1 = b_2 + p_2 \times y_2$

有解的充要条件$(b_1 - b_2)\ mod\ gcd(p_1, p_2) = 0$

设$x = p_1 \times y_1 + b_1$，若存在x满足二式，

则带入x：$p_1 \times y_1 + b_1 = b_2 \pmod{p_2}$，

所以$y_1 = \frac{b_2 - b_1}{p_1} \pmod{p_2}$，

该式有解当且仅当$gcd(p_1, p_2) | (b_2 - b_1)$，

式$p_1 \times y_1 + p_2 \times y_2 = b_2 - b_1$成立。

扩展中国剩余定理-算法实现

关于通解：

所有的$x \bmod \mathrm{lcm}(p_1,p_2)$有唯一解，这样便可通过特解求通解$x' = x + k \times \mathrm{lcm}(p_1..p_m)$。

对于更多的同余方程，先联立两个方程，可得$x \equiv x'(\bmod\ \mathrm{lcm}(p_1\,,p_2))$。

令$P' = \mathrm{lcm}(p_1,p_2)$，即可建立新方程$x \equiv x'(\bmod\ P')$，继续联立便可求得最终的特解。

部分代码如下：

```
int d=b2-b1,g=exgcd(p1,p2),p=p2/g;
if (d%g==0) {
 x=((x*d/g)%p+p)%p,b1=x*p1+b1,p1=(p1*p2)/g;
 return 1;
}
return 0;
```

屠龙勇士(NOI2018)

小 D 最近在网上发现了一款小游戏。游戏的规则如下：

- 游戏的目标是按照编号1→n顺序杀掉n条巨龙，每条巨龙拥有一个初始的生命值a_i。同时每条巨龙拥有恢复能力，当其使用恢复能力时，它的生命值就会每次增加p_i，直至生命值非负。只有在攻击结束后且当生命值 恰好为0时它才会死去。
- 游戏开始时玩家拥有m把攻击力已知的剑，每次面对巨龙时，玩家只能选择一把剑，当杀死巨龙后这把剑就会消失，但作为奖励，玩家会获得全新的一把剑。

小D觉得这款游戏十分无聊，但最快通关的玩家可以获得ION2018的参赛资格，于是小D决定写一个笨笨的机器人帮她通关这款游戏，她写的机器人遵循以下规则：

- 每次面对巨龙时，机器人会选择当前拥有的，攻击力不高于巨龙初始生命值中攻击力最大的一把剑作为武器。如果没有这样的剑，则选择 攻击力最低 的一把剑作为武器。
- 机器人面对每条巨龙，它都会使用上一步中选择的剑攻击巨龙固定的 x次，使巨龙的生命值减少x×ATK。
- 之后，巨龙会不断使用恢复能力，每次恢复p_i生命值。若在使用恢复能力前或某一次恢复后其生命值为0，则巨龙死亡，玩家通过本关。

那么显然机器人的攻击次数是决定能否最快通关这款游戏的关键。小D现在得知了每条巨龙的所有属性，她想考考你，你知道应该将机器人的攻击次数x设置为多少，才能用最少的攻击次数通关游戏吗？

当然如果无论设置成多少都无法通关游戏，输出-1即可。

$T<=5, n<=10^5, m<=10^5, lcm(p_i)<=10^{12}, a_i<=10^{12}, ATK<=10^6$

算法：

根据题意，对于任一条巨龙i，必然存在一把唯一确定的剑，其攻击力为A且能杀掉这条巨龙，此时满足：

$$\mathrm{Ax} = a_i + p_i \times y$$

$$\text{即}\mathrm{Ax} \equiv a_i \ (mod\ p_i)$$

首先可以用multiset/BT/ST维护杀掉每一条巨龙对应的剑，这样A_j和a_i、p_i相关方程组可列出：

$$A_j\mathrm{x} \equiv a_i \ (mod\ p_i)$$

本题就转化为，给定同余方程组：

$$\begin{cases} B_1\mathrm{x} \equiv a_1 \ (mod\ p_1) \\ \dots\dots \\ B_n\mathrm{x} \equiv a_n \ (mod\ p_n) \end{cases}$$

求最小非负整数解。

算法：

对于给定同余方程组：

$$\begin{cases} B_1 x \equiv a_1 \ (mod\ p_1) \\ \cdots\cdots \\ B_n x \equiv a_n \ (mod\ p_n) \end{cases}$$

若已求得前i-1个方程的最小非负整数解ans。

设M=lcm($p_1,p_2,\cdots,p_{i-1}$)，则对于任意整数x，ans+Mx就是前i-1个方程的通解。

若要获得前i个方程的最小非负整数解，可构造出一个式子：

$$B_i(\mathrm{ans} + \mathrm{Mx}) \equiv a_i \ (mod\ p_i)$$

满足该式的x就是前i个方程的解。

移项得：

$$B_i \mathrm{Mx} \equiv a_i - B_i \mathrm{ans} \ (mod\ p_i)$$

即 $B_i \mathrm{Mx} + p_i y \equiv a_i - B_i \mathrm{ans}$

对该方程做exgcd求前i个方程的最小非负整数解ans。

ps：由于相对exCRT有些微变形，这里

$$M' = M \times \frac{p_i}{\gcd(p_i, B_i M)}$$

而不是求lcm。其他小问题就不说了。

谢谢聆听