扩展欧几里得算法

回顾欧几里得算法(详细见第五讲中最大公约数和最小公倍数)

定理: $\forall a,b \in N, b \neq 0, \gcd(a,b)=\gcd(b,a\%b)$

```
int gcd(int a,int b){//最大公约数
    if(b==0)
        return a;
    return gcd(b,a%b);
    //亦可用下行代码解决
    //return b?gcd(b,a%b):a;
}
```

背景问题:已知整数 a,b,n, 问方程 $ax+by=n$ 什么时候有整数解?如何求解所有的整数解?

有解的充分必要条件: $\gcd(a,b)$ **可以整除** n **.**

例如: $4x+6y=8$, $3x+12y=18$ 有整数解, $4x+6y=7$ 没有整数解.

证明: 令 $a=\gcd(a,b)\times a'$, $b=\gcd(a,b)\times b'$,

可得 $ax+by=\gcd(a,b)\times a'\times x+\gcd(a,b)\times b'\times y=\gcd(a,b)[a'x+b'y]=n$.

如果 x,y,a',b' 均为整数, 则 $[a'x+b'y]$ 也为整数, 则 n 必须是 $\gcd(a,b)$ 的倍数, 才有整数解.

求 $ax+by=n$ 的步骤

✓ 判断是否 $ax+by=n$ 有解: $\gcd(a,b)$ **可以整除** n **.**

✓ **求一个特解** (x_0,y_0), 通解;

如何求特解 (x_0,y_0)? ——扩展欧几里得算法

扩展: 裴蜀定理: 若 a,b 是整数, 且 $\gcd(a,b)=d$, 那么对于任意整数 $x,y,ax+by$ 都一定

是 d 的倍数, 特别地, 一定存在整数 x,y , 使得 $ax+by=\gcd(a,b)=d$ 成立.

此处不给出证明　证明链接

$ax+by=\gcd(a,b)$ 过程:

- 当 $b=0$ 时, $ax+by=\gcd(a,b)=\gcd(a,0)=a$, 故 $x=1,y=0$;
- 当 $b\neq 0$ 时, 已知 $\gcd(a,b)=\gcd(b,a\%b)$ 且 $bx'+(a\%b)y'=\gcd(b,a\%b)$

故 $bx'+(a-\left\lfloor\frac{a}{b}\right\rfloor\times b)y'=\gcd(b,a\%b)$, 将该式展开得:

$$ay'+b(x'-\left\lfloor\frac{a}{b}\right\rfloor\times y')=\gcd(b,a\%b)=\gcd(a,b)$$

故 $x=y',y=x'-\left\lfloor\frac{a}{b}\right\rfloor\times y'$

然后采取递归算法, 先求出下一层的 x' 和 y' , 再利用公式回带计算。

```
int exgcd(int a,int b,int &x,int &y){//扩展欧几里得算法,x,y 为C++的引用
    if(b==0){
        x=1,y=0;
        return a;
    }
    int d=exgcd(b,a%b,x,y);
    int temp=x;
    x=y;
    y=temp-a/b*y;
    return d;
}
```

如何求 $ax+by=n$ 的一个特解

- ✓ 判断是否 $ax+by=n$ 有解;
- ✓ 用扩展欧几里得算法**求** $ax+by=\gcd(a,b)$ **的特解** (x'',y'') ;
- ✓ 在 $ax+by=\gcd(a,b)$ 的两边同时乘以 $\frac{n}{\gcd(a,b)}$, 得到:

$$ax''\frac{n}{\gcd(a,b)}+by''\frac{n}{\gcd(a,b)}=n;$$

✓ 对照 $ax+by=n$，一个特解为 $\begin{cases} x_0 = x''\dfrac{n}{\gcd(a,b)} \\ y_0 = y''\dfrac{n}{\gcd(a,b)} \end{cases}$.

拓展: $ax+by=n$ 通解为: $\begin{cases} x = x_0 + \dfrac{b}{\gcd(a,b)} \times t \\ y = y_0 - \dfrac{a}{\gcd(a,b)} \times t \end{cases}$, t为任意整数

同余逆元

1. 同余

设m是正整数，若a,b是整数，且$m|(a-b)$，则称a和b模m同余. 也就是说，a除以m得到的余数，和b除以m的余数相同；或者说，$a-b$除以m，余数为0.（$a-b$**是**m**的整数倍**）

把a和b模m同余记为$a\equiv b(\bmod m)$，m称为同余的模.

举例：

✧ 由于$7|(18-4)$，所以$18\equiv 4(\bmod 7)$，18除以7余数是4，4除以7余数也是4；

✧ $3\equiv -6(\bmod 9)$，3除以9余数是3，-6除以9的余数也是3；

✧ 13和5模9不同余，由于13除以9余数是4，5除以9余数是5.

性质及定理

✓ 若a和b是整数，m为正整数，则$a\equiv b(\bmod m)$当且仅当$a\bmod m=b\bmod m$；

✓ 把**同余式转化为等式**。若a和b是整数，则$a\equiv b(\bmod m)$当且仅当存在整数，使得$a=b+km$. 例如：$19\equiv -2(\bmod 7)$，有$19=-2+3\times 7$.

✓ 设m为正整数，模m的同余满足下面的性质：
  - ❖ 自反性. 若a是整数，$a\equiv a(\bmod m)$；
  - ❖ 对称性. 若a和b是整数，且$a\equiv b(\bmod m)$，则$b\equiv a(\bmod m)$；
  - ❖ 传递性. 若a,b,c是整数，且$a\equiv b(\bmod m)$和$b\equiv c(\bmod m)$，则$a\equiv c(\bmod m)$.

一元线性同余方程$ax\equiv b(\bmod m)$，a,b,m都是整数，求x.

✓ ax除以m与b除以m两者余数相同

✓ $ax-b$是m的整数倍，设倍数为y，那么$ax-b=my$，移项得$ax-my=b$，y可以是负数，改写为$ax+my=b$，这就是**扩展欧几里得算法**的二元一次不定方程（当且仅

当 $\gcd(a,m)$ 能整除 b 时, 有整数解).

2. 逆

求解一般形式的同余方程 $ax \equiv b(\bmod m)$, 需要用到逆。

定义:

给定整数 a, 且满足 $\gcd(a,m)=1$, 称 $ax \equiv 1(\bmod m)$ 的一个解为 a 模 m 的逆, 记为 a^{-1}.

例如: $8x \equiv 1(\bmod 31)$, 有一个解是 $x=4$, 4 是 8 模 31 的逆. 所有的解, 例如 35、66 等都是 8 模 31 的逆.

求法:

1. 扩展欧几里得算法求单个逆

$ax \equiv 1(\bmod m)$ 转化为 $ax+my=1$, 先用扩展欧几里得求出 $ax+my=1$ 的一个特解 x_0, 通解为 $x = x_0 + tm, t$为任意整数, 然后通过取模操作算出最小整数解 $((x_0 \bmod m)+m)(\bmod m)$, 原因求出来的 x_0 可能是负数而且可能不是最小正整数.

```
typedef long long ll;

ll exgcd(ll a,ll b,ll &x,ll &y){//扩展欧几里得算法,x,y 为C++的引用
    if(b==0){
        x=1,y=0;
        return a;
    }
    ll d=exgcd(b,a%b,x,y);
    ll temp=x;
    x=y;
    y=temp-a/b*y;
    return d;
}

ll inv(ll a,ll p){//求逆元
    ll x,y;
    if(exgcd(a,p,x,y)!=1)//无解
```

```
        return -1;
    return (x%p+p)%p;
}
```

2. 费马小定理求单个逆

费马小定理(Fermat’s little theorem)是数论中的一个重要定理,在1636年提出.**如果 p 是一个质数,而整数 a 不是 p 的倍数,则有 $a^{p-1} \equiv 1 \pmod p$. 证明略**

$a^{p-1} = a \times a^{p-2} \equiv 1 \pmod p$,那么 $a^{p-2} \bmod p$ 就是 a 模 p 的逆,故此处使用**快速幂**即可.

```
ll fast_pow(ll a,ll b,ll p){//快速幂
    ll res=1;
    while(b){
        if(b%2==1)//b&1
            res=res*a%p;
        a=a*a%p;
        b/=2;//b>>=1;
    }
    return res;
}

ll inv(ll a,ll p){
    return fast_pow(a,p-2,p);
}
```

欧拉函数

$1\sim N$ 中与 N 互质的数的个数被称为欧拉函数，记为 $\phi(N)$.

若在算术基本定理中，$N=p_1^{c_1}p_2^{c2}\cdots p_m^{c_m}$，则：

$$\phi(N)=N\times\frac{p_1-1}{p_1}\times\frac{p_2-1}{p_2}\times\cdots\times\frac{p_m-1}{p_m}=N\times\prod_{\text{质数}p|N}(1-\frac{1}{p})$$

例如：$100=2^2\times5^2$，$\phi(100)=100\times\frac{1}{2}\times\frac{4}{5}=40$

小于 100 且与 100 互质的数：$\begin{matrix}1,3,7,9,11,13,17,19,21,23,\\27,29,31,33,37,39,41,43,47,49,\\51,53,57,59,61,63,67,69,71,73,\\77,79,81,83,87,89,91,93,97,99\end{matrix}$（每行 10 个）

同理：$\phi(10)=10\times\frac{1}{2}\times\frac{4}{5}=4$，$\phi(30)=30\times\frac{1}{2}\times\frac{2}{3}\times\frac{4}{5}=8$，$\phi(49)=49\times\frac{6}{7}=42$.

证明：基于**容斥原理**

设 p 是 N 的质因子，$1\sim N$ 中 p 的倍数有 $p,2p,3p,\cdots,\left\lfloor\frac{N}{p}\right\rfloor p$，共 $\left\lfloor\frac{N}{p}\right\rfloor$ 个. 同理若 q 也是 N 的质因子，$1\sim N$ 中 q 的倍数有 $q,2q,3q,\cdots,\left\lfloor\frac{N}{q}\right\rfloor q$，共 $\left\lfloor\frac{N}{q}\right\rfloor$ 个. 如果我们把这 $\left\lfloor\frac{N}{p}\right\rfloor+\left\lfloor\frac{N}{q}\right\rfloor$ 个数去掉，那么 $p\times q$ 的倍数被排除了两次，需要加回来一次. 因此，$1\sim N$ 中不与 N 含有共同质因子 p 或 q 的个数为：

$$N-\left\lfloor\frac{N}{p}\right\rfloor-\left\lfloor\frac{N}{q}\right\rfloor+\left\lfloor\frac{N}{pq}\right\rfloor=N(1-\frac{1}{p}-\frac{1}{q}+\frac{1}{pq})$$

而 $1-\frac{1}{p}-\frac{1}{q}+\frac{1}{pq}=1-\frac{1}{p}-\frac{1}{q}(1-\frac{1}{p})=(1-\frac{1}{p})(1-\frac{1}{q})$，故上式得到 $N(1-\frac{1}{p})(1-\frac{1}{q})$.

实际上，上述思想被称为容斥原理. 类似的，可以在 N 的全部质因子使用容斥原理，即可得到 $1\sim N$ 中不与 N 含有任何共同质因子的个数，也就是与 N 互质的个数.

```
int phi(int x){
    int ans=x;
    for(int i=2;i<=x/i;i++)
        if(x%i==0){
            ans=ans/i*(i-1);
            while(x%i==0)
                x/=i;
        }
    if(x>1)
        ans=ans/x*(x-1);//注意此处不用1-1/p
    return ans;
}
```