原创 计算同余方程和同余方程组（中国剩余定理）

2019-05-04 22:35:12 _-Y-_-Y-_ 阅读数 78 更多 编辑

同余

定义1.1 如果$(a-b)mod\ \ m=0$，则称a和b模m同余，记为$a\equiv b(mod\ \ m)$。（a，b为整数，m为正整数）

定理1.1 $(a-b)mod\ \ m=0$当且仅当存在整数k，$a=b+km$。（a，b为整数，m为正整数）

定理1.2 给出一个正整数m和三个整数a,b和c，$a\equiv b(mod\ \ m)$。则
$a+c\equiv b+c(mod\ \ m)$；
$a-c\equiv b-c(mod\ \ m)$；
$ac\equiv bc(mod\ \ m)$。
注意：同余式两边同时除以一个整数并不一定保持同余 例如$10\equiv 4(mod\ \ 6)$同时除以2则不同余

推论1.1 $GCD(c,m)=1$并且$ac\equiv bc(mod\ \ m)$时$a\equiv b(mod\ \ m)$（a，b 为整数，m，c 为正整数）

推论1.2 如果$ad\equiv bd(mod\ \ md)$，则$a\equiv b(mod\ \ m)$（a，b 为整数，m，d 为正整数）

定理1.3 若$d=GCD(c,m)$并且$ac\equiv bc(mod\ \ m)$可以推出$a\equiv b(mod\ \ (\frac{m}{d}))$

证明 定理1.3 $ac\equiv bc(mod\ \ m)$可以写成$c(a-b)mod\ \ m=0$即$c(a-b)=km$。令$c(a-b)DIV\,d=kmDIV\,d$。
又因为$d=GCD(c,m)$可以推出$GCD(\frac{c}{d},\frac{m}{d})=1$
所以$(a-b)mod\ \ (\frac{m}{d})=0$即$a\equiv b(mod\ \ (mDIV\,d))$

模运算规则如下

$(a+b)\%p=(a\%p+b\%p)\%p$
$(a–b)\%p=(a\%p–b\%p)\%p$
$(a*b)\%p=(a\%p*b\%p)\%p$
$(a^b)\%p=((a\%p)^b)\%p$
结合律：
$((a+b)\%p+c)\%p=(a+(b+c)\%p)\%p$
$((a*b)\%p*c)\%p=(a*(b*c)\%p)\%p$
交换律：
$(a+b)\%p=(b+a)\%p$
$(a*b)\%p=(b*a)\%p$
分配律：
$((a+b)\%p*c)\%p=((a*c)\%p+(b*c)\%p)\%p$

例1

https://cn.vjudge.net/problem/POJ-1995

AC code

```
#include <cstdio>
#include <algorithm>
using namespace std;
typedef long long ll;
ll pow(ll x,ll n,ll mod)//快速幂
{
    ll res=1;
    while(n>0)
    {
       if(n%2==1)
       {
         res=res*x;
         res=res%mod;
       }
       x=x*x;
       x=x%mod;
```

```
            n>>=1;
        }
        return res;
}
int main(){
        ll z,m,h,a,b;
        scanf("%lld", &z);
        while(z--){
            ll ans=0;
            scanf("%lld %lld", &m, &h);
            for(int i=0;i<h;i++){
                scanf("%lld %lld", &a, &b);
                ans=(ans+pow(a,b,m))%m;
            }
            printf("%lld\n", ans);
        }
        return 0;
}

```

一元线性同余方程

定义2.1 形如 $ax \equiv b(mod\ \ m)$ 的同余式被称为一元线性同余方程 （$a，\ b$为整数，$m，$为正整数，x为未知整数）
定理2.1 令$GCD(a,m)=d$如果$b\ mod\ \ d!=0$，则$ax \equiv b(mod\ \ m)$无解。
如果$bmod\ \ d=0$，则$ax \equiv b(mod\ \ m)$恰有d个模m不同余的解。
证明 定理2.1 如果$ax \equiv b(mod\ \ m)$，$ax=b+ym$。即$ax-ym=b$。$(exgcd)(x,y$为整数$)$
如果$bmod\ \ d!=0$，则$ax-ym=b$无解；如果$bmod\ \ d=0$，则$ax-ym=b$有无穷解：
$x=x0+k*(\frac{m}{d})$，$y=y0+k*(\frac{a}{d})$。
设$x1=x0+k1*(\frac{m}{d})$，$x2=x0+k2*(\frac{m}{d})$，如果$x1 \equiv x2(mod\ \ m)$，
则$k1*(\frac{m}{d}) \equiv k2*(\frac{m}{d})(mod\ \ m)$
又因为$GCD(\frac{m}{d},m)=\frac{m}{d}$，可得$k1 \equiv k2(mod\ \ d)$。
因此$ax-ym=b$的不同余的解的集合可以通过$x=x0+k*(\frac{m}{d})$得到，其中k为$0,1,...,(d-1)$。
推论 2.1 如果GCD（$a，\ m$）$=1$，则一次同余式$ax+b \equiv 0(mod\ \ m)$有解
定理2.2 $ax \equiv b(mod\ \ m)$，计算x的算法如下
计算$d=GCD(a,m)$和$d=ax'+my'$的解(x',y')，其中x'是$ax' \equiv d(mod\ \ m)$的解。
如果$bmod\ \ d!=0$，则$ax \equiv b(mod\ \ m)$无解
否则存在d个模m不同余的解其中$x0=x'*(\frac{b}{d})mod\ \ m$。
$xi=(x0+i*(\frac{m}{d}))mod\ \ m$，$1 \le i \le d-1$

例2

https://cn.vjudge.net/problem/POJ-2115

AC code

```
/*
在这个题中，b=B-A , a=c, m=2^k
*/
#include <cstdio>
#include <algorithm>
using namespace std;
typedef long long ll;
ll exgcd(ll a,ll b,ll &x, ll &y){
  if(b==0){ x=1,y=0;return a;}
  ll t=exgcd(b,a%b,x,y);
  ll x0=x,y0=y;
  x=y0,y=x0-(a/b)*y0;
  return t;
}
ll mabs(ll x) {return x>0?x:-x;}
int main(){
    ll aa,bb,cc,pp;
    while(~scanf("%lld %lld %lld %lld", &aa, &bb, &cc, &pp),(aa+bb+cc+pp)){
        ll a=cc,m=(ll)1<<pp,b=bb-aa;
        ll d,x,y;
        d=exgcd(a,m,x,y);
        if(b%d!=0) printf("FOREVER\n");
        else{
```

```
            x=(x*(b/d))%m;
            ll t=mabs(m/d);
            x=(x%t+t)%t;
            printf("%lld\n", x);
        }
    }
    return 0;
}

```

例3

https://cn.vjudge.net/problem/ZOJ-3609

AC code

```
#include <bits/stdc++.h>
using namespace std;
int exgcd(int a,int b,int &x, int &y){
  if(b==0){ x=1,y=0;return a;}
  int t=exgcd(b,a%b,x,y);
  int x0=x,y0=y;
  x=y0,y=x0-(a/b)*y0;
  return t;
}
int main(){
  int T;
  int a,b=1,m,d,x,y;
  scanf("%d", &T);
  while(T--){
    scanf("%d %d", &a, &m);
    d=exgcd(a,m,x,y);
    if(b%d==1){
      printf("Not Exist\n");
      continue;
    }
    x=x*(b/d)%m;
    int t=m/d;
    x=(x%t+t)%t;
    if(x==0) x+=t;
    printf("%d\n", x);
  }
  return 0;
}

```

同余方程组

定理 3.1（中国剩余定理） 设$n1, n2, n3....nk$是两两互素的正整数，则同余方程组

$a \equiv a_1 (\mathrm{mod}\ \ n_1)$

$a \equiv a_2 (\mathrm{mod}\ \ n_2)$

....

....

....

$a \equiv a_k (\mathrm{mod}\ \ n_k)$

有模$n = n_1 \times n_2 \times n_3 ... n_k$唯一解
即$a = (a1 \times c_1 + ... + a_i \times c_i + ... + a_k \times c_k) mod(n_1 \times n_2 \times ... \times n_k)$。

证明 定理 3.1 设$m_i = \frac{n}{n_i}(1 \le i \le k) gcd(n_i, m_i) = 1$则存在整数$n_i'$和$m_i'$使得$m_i * m_i' + n_i * n_i' = 1$即$m_i * m_i' \equiv 1(mod\ \ n_i)$
又因为$GCD(n_i, n_j) = 1$并且$m_i = \frac{n}{n_i}$则$m_j\ mod\ \ n_i = 0(i! = j)$

所以$a_j * m_j * m_j' = 0\ (mod\ \ n_i)(i, j = 1, 2, 3...k, i! = j)$
$a_1 m_1 m_1' + a_2 m_2 m_2' + a_3 m_3 m_3' + ... + a_k m_k m_k' \equiv a_i m_i m_i'(mod\ \ n_i)(i = 1, 2...k)$
所以$a = (a_1 * c_1 + \ldots + a_i * c_i + \ldots + a_k * c_k) mod(n_1 \times n_2 \times \ldots \times n_k)$。

应用 计算同余方程组

（1）先计算 $m_i \quad m_i = \frac{n}{n_i}$

（2）再通过 同余方程 $m_i * m_i' = 1(mod\ \ n_i)$ 或者 利用扩展欧几里德算法$n_i * x + m_i * y = 1$计算m_i'

（3）计算 $c_i = m_i * m_i'$

（4） 计算 $a = (a_1 \times c_1 + \ldots + a_i \times c_i + \ldots + a_k \times c_k) mod(n_1 \times n_2 \times \ldots \times n_k)$。

例4

https://cn.vjudge.net/problem/POJ-1006

AC code

```
#include <cstdio>
using namespace std;
int exgcd(int a,int b,int &x, int &y){
  if(b==0){ x=1,y=0;return a;}
  int t=exgcd(b,a%b,x,y);
  int x0=x,y0=y;
  x=y0,y=x0-(a/b)*y0;
  return t;
}
int ty(int a,int m){
  int b=1,d,x,y;
  d=exgcd(a,m,x,y);
  if(b%d==1){
    return -1;
  }
  x=x*(b/d)%m;
  int t=m/d;
  x=(x%t+t)%t;
  if(x==0) x+=t;
  return x;
}
int main(){
  int x,y;
  int m1=28*33,m2=23*33,m3=23*28;
  int n1=23,n2=28,n3=33,n=23*28*33;
  int c1,c2,c3;
  int p,e,i,d;
  c1=ty(m1,n1)*m1;
  c2=ty(m2,n2)*m2;
  c3=ty(m3,n3)*m3;
  int num=0;
  while(~scanf("%d %d %d %d", &p, &e, &i, &d)){
    num++;
    if(p==-1&&e==-1&&i==-1&&d==-1) break;
    int ans=(c1*p+c2*e+c3*i-d+n)%n;
    if(ans==0) ans=n;
    printf("Case %d: the next triple peak occurs in %d days.\n", num, ans);
  }
  return 0;
}

```

有 0 个人打赏　　　　文章最后发布于: 2019-05-04 22:35:12