扩展中国剩余定理

例题

【模板】扩展中国剩余定理

题意：求解以下同余方程组

$$ \left\{\begin{array}{ccc} x &\equiv&a_1\pmod{m_1}\\ x &\equiv&a_2\pmod{m_2}\\ &\vdots&\\ x &\equiv& a_n\pmod{m_n}\\ \end{array}\right. $$

不保证 m_i 互质，保证有解

题解

对于只有 2 个方程的情况$x\equiv a_1\pmod{m_1};x\equiv a_2\pmod{m_2}$等价于 $x=a_1+m_1t_1=a_2+m_2t_2$即 $m_1t_1-m_2t_2=a_2-a_1$用扩展欧几里得解出 t_1若无解则方程组无解），从而得到 x 的解 $x\equiv x_0\pmod{\text{lcm}(m_1,m_2)}$从而将两个方程合并为一个$n$ 个方程即执行 $n-1$ 次扩展欧几里得算法，不断合并方程，直至得到最终解。

代码

```
#include<bits/stdc++.h>
using namespace std;
const int N=1e5+5;
typedef long long ll;
ll m[N],a[N];
ll gcd(ll x,ll y){
    return !y?x:gcd(y,x%y);
}
ll lcm(ll x,ll y){
    return y/gcd(x,y)*x;
}
void exgcd(ll a,ll b,ll &x,ll &y){
    if(!b){
        x=1,y=0;return;
    }
    exgcd(b,a%b,y,x);
    y-=x*(a/b);
}
ll mul(ll x,ll y,ll mod){
    ll ans=0;
    x%=mod;y%=mod;
    while(y){
        if(y&1) ans=(ans+x)%mod;
        x=(x+x)%mod;
        y>>=1;
    }
    return ans;
```

```
}
ll calc(ll M,ll mi,ll c,ll x0){
    ll g=gcd(M,mi);
    ll x,y;
    exgcd(M,mi,x,y);
    ll temp=lcm(M,mi);
    c=(c%mi+mi)%mi;
    ll ret=mul(x,c/g,mi);
    ret=(ret%mi+mi)%mi;
    return (mul(ret,M,temp)+x0%temp)%temp;
}
int main(){
    int n;
    scanf("%d",&n);
    for(int i=1;i<=n;i++) scanf("%lld %lld",&m[i],&a[i]);
    ll M=m[1];
    ll ans=a[1]%M;
    for(int i=2;i<=n;i++){
        ans=calc(M,m[i],a[i]-ans,ans);
        M=lcm(M,m[i]);
        ans=(ans+M)%M;
    }
    printf("%lld",ans);
    return 0;
}
```