Miller Rabin - 判断是否为素数 O(klog^3n)

inline ll qmul(ll a,ll b,ll mod){

return (ll)(\_\_int128(a)\*b%mod);

}

ll ksm(ll a,ll n,ll mod){

ll ans=1;

while(n){

if(n&1) ans=qmul(ans,a,mod);

a=qmul(a,a,mod);

n>>=1;

}return ans;

}

ll d[8]={2,3,5,7,11,13,79,97};

bool fun(ll n){

if(n<3) return n==2;

if(n%2==0) return 0;

ll p=n-1,c=0;

while(p%2==0) p>>=1,c++;

for(int i=0;i<8;i++){

if(d[i]==n) return 1;

ll x=ksm(d[i],p,n),y=x;

for(int j=0;j<c;j++){

x=qmul(x,x,n);

if(x==1&&!(y==1||y==n-1)) return 0;

y=x;

}

if(x!=1) return 0;

}return 1;

}

Pollard-Rho – 大数质因数分解 O(n^(1/4))

ll gcd(ll a,ll b){

return !b?a:gcd(b,a%b);

}

inline ll qmul(ll a,ll b,ll mod){

return (ll)(\_\_int128(a)\*b%mod);

}

ll ksm(ll a,ll n,ll mod){

ll ans=1;

while(n){

if(n&1) ans=qmul(ans,a,mod);

a=qmul(a,a,mod);

n>>=1;

}return ans;

}

ll d[8]={2,3,5,7,11,13,79,97};

bool fun(ll n){

if(n<3) return n==2;

if(n%2==0) return 0;

ll p=n-1,c=0;

while(p%2==0) p>>=1,c++;

for(int i=0;i<8;i++){

if(d[i]==n) return 1;

ll x=ksm(d[i],p,n),y=x;

for(int j=0;j<c;j++){

x=qmul(x,x,n);

if(x==1&&!(y==1||y==n-1)) return 0;

y=x;

}

if(x!=1) return 0;

}return 1;

}

inline ll f(ll x,ll c,ll m) //生成随机数列

{

return (\_\_int128(x)\*x+c)%m;

}

ll PR(ll x) //随机找到x的一个因数

{

ll s=0,t=0,c=1ll\*rand()%(x-1)+1;

int stp=0,goal=1;

ll val=1;

for(goal=1;;goal<<=1,s=t,val=1)

{

for(stp=1;stp<=goal;++stp)

{

t=f(t,c,x);

val=\_\_int128(val)\*abs(t-s)%x;

if((stp%127)==0)

{

ll d=gcd(val,x);

if(d>1)

return d;

}

}

ll d=gcd(val,x);

if(d>1)

return d;

}

}

ll mx;

inline void fac(ll x) //递归找x的最大因数

{

if(x<=mx || x<2)

return;

if(fun(x))

{

mx=mx>x?mx:x;

return;

}

ll p=x;

while(p>=x)

p=PR(x);

while((x%p)==0)

x/=p;

fac(x),fac(p);

}

vector<pair<ll,int> >b; //b存放质因数及对应的个数

while(1){

mx=0;

fac(x);int c=0;

while((x%mx)==0){

c++,x/=mx;

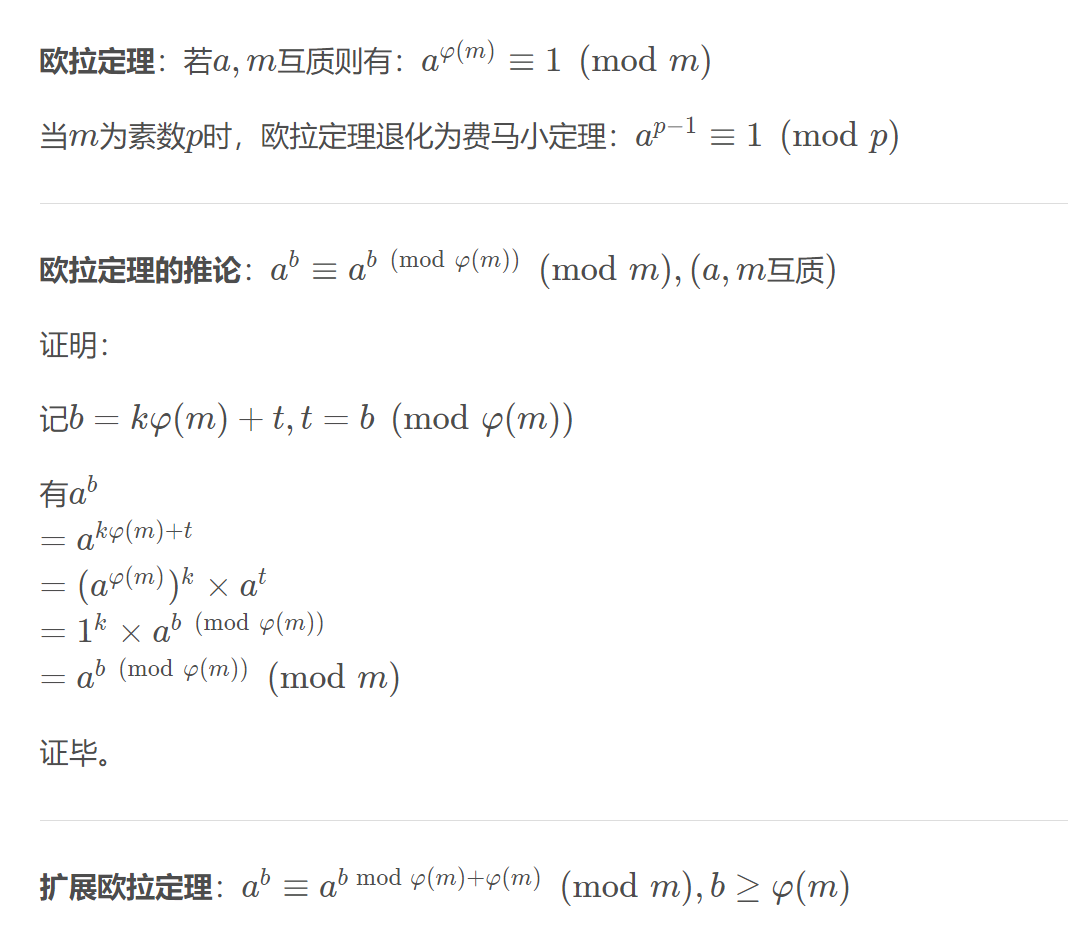
}

b.pb({mx,c});

if(x==1) break;

}

欧拉定理及扩展欧拉定理



#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

const int N=1e3+5,M=2e4+5,inf=0x3f3f3f3f,mod=1e9+7;

#define mst(a,b) memset(a,b,sizeof a)

#define PII pair<int,int>

#define fi first

#define se second

#define pb push\_back

bool f;

inline void read(ll &s){

int w=1;char c;

while(c=getchar(),!isdigit(c)){

if(c=='-') w=-1;

}

while(isdigit(c)) s=(s<<3)+(s<<1)+(c&15),c=getchar();

s\*=w;

}

inline ll readm(ll m){

ll s=0;char c;while(c=getchar(),!isdigit(c)) ;

while(isdigit(c)){

s=(s<<3)+(s<<1)+(c&15);

if(s>=m){

s%=m;

f=true;

}

c=getchar();

}

return s;

}

ll phi(ll n){

ll s=n;

for(ll i=2;i\*i<=n;i++){

if(n%i==0){

s-=s/i;

while(n%i==0) n/=i;

}

}

if(n>1) s-=s/n;return s;

}

ll qmul(ll x,ll y,ll m){ //龟速乘 比x\*y慢，但是能在取模意义下获得正确值

ll s=0;

while(y){

if(y&1) s=(s+x)%m;

x=(x+x)%m;

y>>=1;

}

return s;

}

ll a,m,b;

ll ksm(ll a,ll n,ll m){

ll ans=1;

while(n){

if(n&1) ans=qmul(ans,a,m);

a=qmul(a,a,m);

n>>=1;

}

return ans;

}

int main(){

read(a),read(m);ll phim=phi(m);b=readm(phim);

printf("%lld\n",ksm(a,b+(f?phim:0),m));

return 0;

}