欧拉函数

f[1]=1;

for(int i=2;i<=1000000;i++){

if(!f[i]){

for(int j=i;j<=1000000;j+=i){

if(!f[j])f[j]=j;

f[j]=f[j]/i\*(i-1);

}

}

}//线性筛O(n)

ll ol(ll x){

ll res=x;

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

if(!(x%i)){

res/=i;res\*=i-1;

while(!(x%i))x/=i;

}

}

if(x^1)res/=x,res\*=x-1;

return res;

}//求单个欧拉函数

欧拉降幂//a^a^a^a^…^a降幂

ll tmp=a,ans=0;

if(m==1)return 0;

if(m==2)return a%2;

else {

if(b==0)return 1;

if(b==1)return a%m;

//printf("%lld\n",tmp);

for(int i=1;i<b-1;i++){

ans=tmp;

for(int j=1;j<a;j++){

tmp=tmp\*ans;

//printf("%lld\n",tmp);

if(tmp<f[m])continue;

return quick(a,q(a,b-1,f[m])+f[m],m);

}

}

return quick(a,tmp,m);

}

中国剩余定理解同余方程组

typedef \_\_int128 ll;

inline ll exgcd(ll p,ll q,ll &x,ll &y){

if(!q){

x=1;

y=0;

return p;

}

ll g=exgcd(q,p%q,y,x);

y-=(p/q)\*x;

return g;

}

ll excrt(){

M=b[1];

ans=a[1];

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

ll c=((a[i]-ans)%b[i]+b[i])%b[i];

ll g=exgcd(M,b[i],x,y);

ll p=b[i]/g;

if(c%g!=0)return -1;//判断无解

if(M>ll(1e15))continue;

x=mul(x,c/g,p);//快速乘

ans+=x\*M;

M\*=p;

ans=(ans%M+M)%M;

}

return ans;

}

模拟退火

inline void fire(){//T初始温度 delta温度变化量

double x=ansx,y=ansy;//较优解横坐标和纵坐标

double T=max\_temperature,delta=change\_temperature;

while (T>=min\_temperature){

register double xx=ansx+(rand()\*2-RAND\_MAX)\*T;

register double yy=ansy+(rand()\*2-RAND\_MAX)\*T;//随机横坐标和纵坐标

register double nowans=calc(xx,yy);

register double DE=nowans-ans;

if (DE<0){

ans=nowans;

x=ansx=xx;y=ansy=yy;

}

else if (exp(-DE/T)\*RAND\_MAX>rand()) x=xx,y=yy;

T\*=delta;

}

}