快速数论变换(NTT)

原理

OI Wiki-快速数论变换(NTT)

例题

例 1

题意

P3803 【模板】多项式乘法（FFT）

给定一个 n 次多项式 $F(x)$，和一个 m 次多项式 $G(x)$，求出 $F(x)$ 和 $G(x)$ 的卷积。$1\le n,m\le 10^6$

题解

找到模数 p，使得 $p=qn+1,(n=2^k)$，对于模 p 的原根 g，$\displaystyle g_n\equiv g^q\equiv g^{\frac{p-1}{n}}$，可以看作 FFT 中 n 次单位根 ω_n 的等价，因为它们都是各自所在群的生成元。所以 NTT 的代码只需把 FFT 中的 ω_n 都用 $g^{\frac{p-1}{n}}$ 替换掉就好了。

评价

NTT 模板题

代码

```
#include<bits/stdc++.h>
using namespace std;
const int N=3*1e6+10,P=998244353,G=3,Gi=332748118;//原根 3 及其逆元
typedef long long ll;
int rev[N];
ll a[N],b[N];
ll fastpow(ll x,ll y){//快速幂
    ll ret=1;
    for(;y;y>>=1,x=x*x%P)
        if(y&1) ret=ret*x%P;
    return ret;
}
void NTT(ll *y,int len,int on){//与 FFT 的代码类似
```

```
    for(int i=0;i<len;i++)
        if(i<rev[i])
            swap(y[i],y[rev[i]]);
    for(int mid=1;mid<len;mid<<=1){
        ll wn=fastpow(on==1?G:Gi,(P-1)/(mid<<1));
        for(int j=0;j<len;j+=(mid<<1)){
            ll w=1;
            for(int k=0;k<mid;k++,w=(w*wn)%P){
                int u=y[j+k],t=w*y[j+k+mid]%P;
                y[j+k]=(u+t)%P;
                y[j+k+mid]=(u-t+P)%P;
            }
        }
    }
}
int main(){
    int n,m;
    scanf("%d%d",&n,&m);
    for(int i=0;i<=n;i++) scanf("%lld",&a[i]),a[i]%=P;
    for(int i=0;i<=m;i++) scanf("%lld",&b[i]),b[i]%=P;
    int len=1,l=0;
    while(len<=n+m) len<<=1,l++;//向上凑成2的幂次
    for(int i=0;i<len;i++) rev[i]=(rev[i>>1]>>1)|((i&1)<<(l-1));//蝴蝶变换
    NTT(a,len,1);
    NTT(b,len,1);
    for(int i=0;i<len;i++) a[i]=(a[i]*b[i])%P;
    NTT(a,len,-1);
    ll inv=fastpow(len,P-2);//最后要除以长度
    for(int i=0;i<=n+m;i++)
        printf("%lld ",(a[i]*inv)%P);

    return 0;
}
```