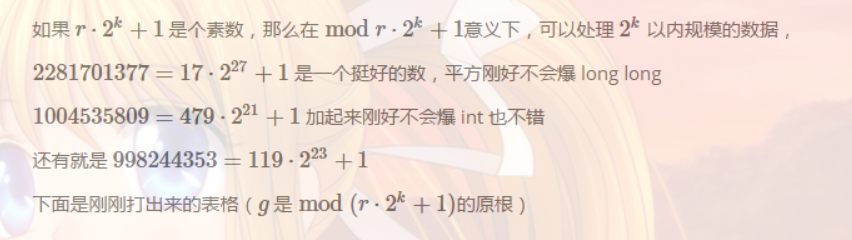
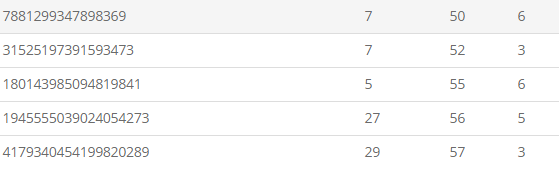
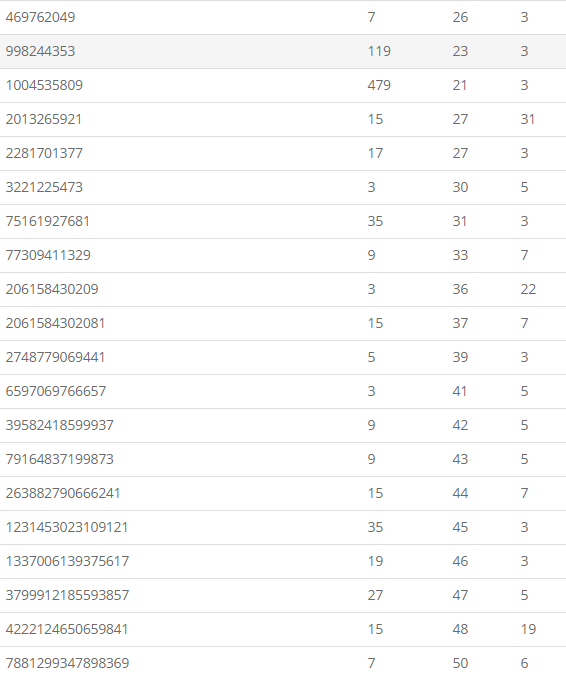
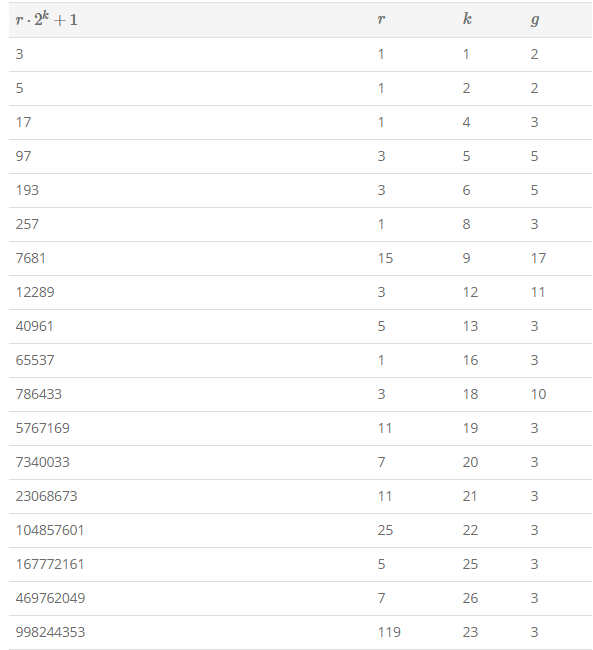
题目同fft，但是，ntt只能解决系数为整数的多项式相乘，其支持模数，但是模数必须是一个特殊值，如下：也就是mod=r\*2^k+1，只有这样才能使用ntt。如果不行的话，就只能使用mtt了



g即为原根（也就是代码中的G），一般我们只要一般情况下只需要记住998244353的原根是3就好，剩下的基本如图所示



普通写法：

#include<bits/stdc++.h>

#define ll long long

#define inf 1e18

#define mn 5000005

using namespace std;

const ll mod=998244353,G=3,Gi=332748118;//这里的Gi是G的除法逆元

const double pi=acos(-1);

ll n,m,res,ans[mn];

ll lim,maxl;//二进制的位数

ll r[mn],a[mn],b[mn];

inline void in(ll &x)

{ll s=0,w=1;char ch=getchar();

while(ch<'0'||ch>'9'){if(ch=='-')w=-1;ch=getchar();}

while(ch>='0'&&ch<='9'){s=s\*10+ch-'0';ch=getchar();}

x=s\*w;

}

ll cal(ll x,ll t)

{ll ret=1;

for(;t;t/=2,x=x\*x%mod)

if(t%2==1)ret=ret\*x%mod;

return ret;

}

void ntt(ll \*a,ll type)

{ll i,mid,k,len,pos;

for(i=0;i<lim;i++)

if(i<r[i])swap(a[i],a[r[i]]);

for(mid=1;mid<lim;mid<<=1)

{

ll wn=cal(G,(mod-1)/(mid\*2));

if(type==-1)wn=cal(wn,mod-2);

for(len=mid<<1,pos=0;pos<lim;pos+=len)

{

ll w=1;

for(k=0;k<mid;k++,w=(w\*wn)%mod)

{

ll x=a[pos+k],y=w\*a[pos+mid+k]%mod;

a[pos+k]=(x+y)%mod;

a[pos+k+mid]=(x-y+mod)%mod;

}

}

}

if(type==-1)

{

ll lim\_inv=cal(lim,mod-2);//N的逆元（N是lim, 指的是2的整数幂）

for(i=0;i<lim;i++)

a[i]=(a[i]\*lim\_inv)%mod;//ntt还是要除以n的，但是这里把除换成逆元了，inv就是n在模p意义下的逆元

}

}

void mul(ll \*a,ll \*b,ll deg)

{ll i;

for(lim=1,maxl=0;lim<=deg;lim<<=1)maxl++;

for(i=0;i<lim;i++)

r[i]=(r[i>>1]>>1)|((i&1)<<(maxl-1));

ntt(a,1);

ntt(b,1);

for(i=0;i<lim;i++)a[i]=a[i]\*b[i]%mod;

ntt(a,-1);

}

int main()

{

ll x,y,z,i,j,k;

while(~scanf("%lld%lld",&n,&m))

{

for(i=0;i<=n;i++){in(a[i]);a[i]=(a[i]+mod)%mod;}//取模好习惯

for(i=0;i<=m;i++){in(b[i]);b[i]=(b[i]+mod)%mod;}

mul(a,b,n+m);

for(i=0;i<=n+m;i++)

printf("%d ",a[i]);

printf("\n");

for(i=0;i<=lim;i++)

a[i]=b[i]=0;

}

return 0;

}

vector打包写法：

#include<bits/stdc++.h>

#define ll long long

#define inf 1e18

#define mn 5000005

using namespace std;

const ll mod=998244353;

inline void in(ll &x)

{ll s=0,w=1;char ch=getchar();

while(ch<'0'||ch>'9'){if(ch=='-')w=-1;ch=getchar();}

while(ch>='0'&&ch<='9'){s=s\*10+ch-'0';ch=getchar();}

x=s\*w;

}

ll cal(ll x,ll t)

{ll ret=1;

for(;t;t/=2,x=x\*x%mod)

if(t%2==1)ret=ret\*x%mod;

return ret;

}

namespace Poly

{

typedef vector<ll>poly;

const ll G=3;

const ll inv\_G=cal(G,mod-2);

ll r[mn],dr[2][22][mn],inv[mn];

void init(const ll t)

{//预处理出来ntt里需要的w和wn，砍掉了一个log的时间

for(ll p=1;p<=t;p++)

{

ll buf1=cal(G,(mod-1)/(1<<p));

ll buf0=cal(inv\_G,(mod-1)/(1<<p));

dr[0][p][0]=dr[1][p][0]=1;

for(ll i=1;i<(1<<p);i++)

{

dr[0][p][i]=dr[0][p][i-1]\*buf0%mod;//逆

dr[1][p][i]=dr[1][p][i-1]\*buf1%mod;

}

}

inv[1]=1;

for(ll i=2;i<=(1<<t);i++)

inv[i]=inv[mod%i]\*(mod-mod/i)%mod;

}

ll ntt\_init(ll n)

{

ll lim=1,maxl=0;

while(lim<n){lim<<=1;maxl++;}

for(ll i=0;i<lim;i++)

r[i]=(r[i>>1]>>1)|((i&1)<<(maxl-1));

return lim;

}

#define ck(x) (x >= mod ? x - mod : x)

void ntt(poly &a,ll type,ll lim)

{

a.resize(lim);

for(ll i=0;i<lim;i++)

if(i<r[i])swap(a[i],a[r[i]]);

for(ll mid=2,j=1;mid<=lim;mid<<=1,j++)

{

ll len=mid>>1;

for(ll pos=0;pos<lim;pos+=mid)

{

ll \*wn=dr[type][j];

for(ll i=pos;i<pos+len;i++,wn++)

{

ll tmp=(\*wn)\*a[i+len]%mod;

a[i+len]=ck(a[i]-tmp+mod);

a[i]=ck(a[i]+tmp);

}

}

}

if(type==0)

{

ll inv\_lim=cal(lim,mod-2);

for(ll i=0;i<lim;i++)

a[i]=a[i]\*inv\_lim%mod;

}

}

poly mul(poly a,poly b)

{

ll deg=a.size()+b.size()-1;

ll lim=ntt\_init(deg);

poly c(lim);

ntt(a,1,lim);

ntt(b,1,lim);

for(ll i=0;i<lim;i++)

c[i]=a[i]\*b[i]%mod;

ntt(c,0,lim);

c.resize(deg);

return c;

}

}

using Poly::poly;

using Poly::mul;

ll n,m,x;

int main()

{

ll x,y,z,i,j,k;

Poly::init(21);

while(~scanf("%lld%lld",&n,&m))

{

poly f,g;

for(i=0;i<=n;i++){in(x);f.push\_back(x+mod%mod);}

for(i=0;i<=m;i++){in(x);g.push\_back(x+mod%mod);}

g=mul(f,g);

for(i=0;i<=n+m;i++)

printf("%lld ",g[i]);

printf("\n");

}

return 0;

}