常系数齐次线性递推

算法思想

求一个满足 k 阶齐次线性递推数列 a_{i} 的第 n 项，即 $a_{n}=\sum_{i=1}^{k} f_{i}×a_{n-i}$ 求 a_{n}

其中 $n≤10^{9},k≤32000,|f_{i}|,|a_{0},...,a_{k-1}|≤10^{9}$

解法是求用快速幂求 x^{n} 对特征多项式 p 取模的结果。

比如求斐波那契数列的第五项 fib_{5} 于是我们相当于 $0x^{0}+0x^{1}+...+1x^{5}$

我们先把 x^{5} 的系数减一，再把 x^{4} 和 x_{3} 系数都加一，从而变成 $0x^{0}+0x^{1}+...+1x^{3}+1x^{4}+0x^{5}$ 相当于对于 $x^{2}-x-1$ 取模。剩下依次类推。

于是原题相当于求出多项式 x^{n} 对多项式 $x_{k}-p_{1}x_{k-1}-p_{2}x_{k-1}-...-p_{k}$ 取模。

算法实现

```
#include <bits/stdc++.h>
#define int long long
using namespace std;
const int N=2000006,GG=3,GI=332748118,mod=998244353;

inline void read(int &X){
    X = 0;
    int w=0; char ch=0;
    while(!isdigit(ch)) {w|=ch=='-';ch=getchar();}
    while(isdigit(ch)) X=(X<<3)+(X<<1)+(ch^48),ch=getchar();
    if (w) X = -X;
}
namespace my_f {
    int qpow(int a,int b,int m=mod) {
        int r=1;
        while(b) {
            if (b&1) r=r*a%m;
            a=a*a%m,b>>=1;
        }
        return r;
    }
    int pgg[30],pgi[30];
    void init() {
        for(register int i=0; i<=23; i++) {
            pgg[i]=qpow(GG,(mod-1)>>i);
            pgi[i]=qpow(GI,(mod-1)>>i);
        }
```

```
    }
    #define ad(x,y) ((x)+(y)>mod?(x)+(y)-mod:(x)+(y))
    #define dc(x,y) ((x)-(y)<0?(x)-(y)+mod:(x)-(y))
    namespace poly {
        int w[N],r[N];
        void NTT(int f[N],int lim,int type) {
            for(register int i=0; i<lim; i++) if (i<r[i])
swap(f[i],f[r[i]]);
            for(register int mid=1,pp=1; mid<lim; mid<<=1,++pp) {
                int Wn=(type==1?pgg[pp]:pgi[pp]);
                w[0]=1;
                for(register int i=1; i<mid; i++) w[i]=w[i-1]*Wn%mod;
                for(register int i=0; i<lim; i+=(mid<<1)) {
                    for(int j=0; j<mid; ++j) {
                        register int y=f[i|mid|j]*w[j]%mod;
                        f[i|mid|j]=dc(f[i|j],y);
                        f[i|j]=ad(f[i|j],y);
                    }
                }
            }
            if (type==-1) {
                int ivv=qpow(lim,mod-2);
                for(register int i=0; i<lim; i++) f[i]=f[i]*ivv%mod;
            }
        }
        int pool[10][N];
        inline void Inv(int f[N],int g[N],int n) { //求逆, pool 0~2
            int *iv=pool[0],*a=pool[1],*b=pool[2];
            for(int i=0; i<=4*n; i++) iv[i]=a[i]=b[i]=0;
            iv[0]=qpow(f[0],mod-2);

            int len=1,lim=1;
            for(len=1; len<=(n<<1); len<<=1) {
                lim=len<<1;

                for(register int i=0; i<=4*len; i++) a[i]=b[i]=0;
                for(register int i=0; i<len; i++) a[i]=f[i],b[i]=iv[i];
                for(register int i=0; i<lim; i++)
r[i]=((r[i>>1]>>1)|((i&1)?len:0));
                NTT(a,lim,1);
                NTT(b,lim,1);
                for(register int i=0; i<lim; i++) a[i]=(2*b[i]-
a[i]*b[i]%mod*b[i]%mod+mod)%mod;
                NTT(a,lim,-1);
                for(int i=0; i<len; i++) iv[i]=a[i];
            }
            for(register int i=0; i<n; i++) g[i]=iv[i];
            for(register int i=n; i<lim; i++) g[i]=0;
        }
        inline void mul(int f[N],int g[N],int n,bool flag=1) // *=, pool
3~4
```

```
        // flag: 是否对n取膜
        {
            int len=1,lim=1;
            while(len<=(n<<1)) len=lim,lim<<=1;
            int *a=pool[3],*b=pool[4];
            for(register int i=0; i<lim; i++) a[i]=b[i]=0;
            for(register int i=0; i<n; i++) a[i]=f[i],b[i]=g[i];
            for(register int i=0; i<lim; i++)
r[i]=((r[i>>1]>>1)|((i&1)?len:0));
            NTT(a,lim,1);
            NTT(b,lim,1);
            for(register int i=0; i<lim; i++) a[i]=a[i]*b[i]%mod;
            NTT(a,lim,-1);
            for(register int i=0; i<2*n; i++) f[i]=a[i];
            for(register int i=2*n; i<=lim; i++) f[i]=0;
            if (flag) for(int i=n; i<2*n; i++) f[i]=0;
        }
        void Mod(int f1[N],int f2[N],int n,int m,int R[N]) //多项式取模, pool
5~6
        //这里只保留了余数,商开到 pool 里而不是传参数修改了
        {
            int *a=pool[5],*b=pool[6],*Q=pool[7];
            for(register int i=0; i<=4*n; i++) a[i]=b[i]=0;
            for(register int i=0; i<n; i++) a[i]=f1[n-1-i];
            for(register int i=n-m+1; i<n; i++) a[i]=0; // a=f1_r%(x^(n-
m+1))
            for(register int i=0; i<m; i++) b[i]=f2[m-1-i];
            for(register int i=n-m+1; i<m; i++) b[i]=0;
            Inv(b,b,n-m+1);
            mul(a,b,n-m+1);
            for(register int i=0; i<=n-m; i++) Q[i]=a[n-m-i];

            for(register int i=0; i<=4*n; i++) a[i]=b[i]=0;
            for(register int i=0; i<n; i++) a[i]=f1[i];
            for(register int i=0; i<m; i++) b[i]=f2[i];
            int len=1,lim=1;
            while(len<=n) len=lim,lim<<=1;
            for(register int i=0; i<lim; i++)
r[i]=((r[i>>1]>>1)|((i&1)?len:0));
            NTT(b,lim,1);
            NTT(Q,lim,1);
            for(register int i=0; i<lim; i++) b[i]=b[i]*Q[i]%mod;
            NTT(b,lim,-1);
            NTT(Q,lim,-1);
            for(register int i=0; i<m-1; i++) R[i]=(a[i]-b[i]%mod+mod)%mod;
            for(register int i=m-1; i<lim; i++) R[i]=0;
        }
        void PowMod(int f[N],int p,int g[N],int n,int h[N]) { // f^p%g, g
有n项,保存在h
            int *res=pool[8];
            for(int i=0; i<=8*n; i++) res[i]=0;
```

```
            res[0]=1; // res=1
            while(p) {
                if (p&1) {
                    mul(res,f,n,0); // res*=f
                    Mod(res,g,2*n,n,res); // res%=g
                }
                mul(f,f,n,0); // f=f*f
                Mod(f,g,2*n,n,f); // f%=g
                p>>=1;
            }
            for(int i=0; i<n-1; i++) h[i]=res[i];
            for(int i=n; i<=8*n; i++) h[i]=0;
        }
    }
    int n,k;
    int a[N],t[N];
    void Input() {
        read(n);read(k);
        for(register int i=1; i<=k; i++)
read(t[i]),t[i]=(t[i]%mod+mod)%mod;
        for(register int i=0; i<k; i++) read(a[i]),a[i]=(a[i]%mod+mod)%mod;
    }
    int f[N],g[N],c[N];
    void Sakuya() {
        init();
        f[1]=1;
        for(register int i=1; i<=k; i++) g[k-i]=(mod-t[i]);
        g[k]=1;
        poly::PowMod(f,n,g,k+1,c);

        int ans=0;
        for(register int i=0; i<k; i++) ans+=a[i]*c[i]%mod;
        ans%=mod;
        printf("%lld\n",ans);
    }

}
#undef int
using namespace my_f;
int main() {
    Input();
    Sakuya();
    return 0;
}
```