# 牛客练习赛68

比赛链接

## D 牛牛的粉丝

### 题意

一个长度为 $n$ 的环，初始位置 $i$ 有 $x_i$ 个人。

接下来每步，每个人有 $\frac as$ 概率向前一步□$\frac bs$ 概率向后一步□$\frac cs$ 概率不动，其中 $s=a+b+c$□

问 $k$ 次后每个位置的人数期望。

### 题解 1

倍增 $\text{dp}$□${dp}(i,j)$ 表示初始只有位置 $0$ 有一人，则 $j$ 步后位置 $i$ 期望有几人。

不难得到状态转移方程，时间复杂度 $O(n^2\log k)$□

```
const int MAXN=505,mod=998244353;
int n,dp[2][MAXN],x[MAXN],ans[MAXN],a,b,c,pos;
int quick_pow(int a,LL b){
    int ans=1;
    while(b){
        if(b&1)ans=1LL*ans*a%mod;
        a=1LL*a*a%mod;
        b>>=1;
    }
    return ans;
}
void Mul(){
    pos=!pos;
    mem(dp[pos],0);
    _for(i,0,n){
        _for(j,0,n)
dp[pos][(i+j)%n]=(dp[pos][(i+j)%n]+1LL*dp[!pos][i]*dp[!pos][j])%mod;
    }
}
void Add(){
    pos=!pos;
    _for(i,0,n){
        dp[pos][i]=0;
        dp[pos][i]=(dp[pos][i]+1LL*a*dp[!pos][(i+n-1)%n])%mod;
        dp[pos][i]=(dp[pos][i]+1LL*b*dp[!pos][(i+1)%n])%mod;
```

```
        dp[pos][i]=(dp[pos][i]+1LL*c*dp[!pos][i%n])%mod;
    }
}
int main()
{
    n=read_int();
    LL k=read_LL(),Bit=0;
    a=read_int(),b=read_int(),c=read_int();
    _for(i,0,n)x[i]=read_int();
    while(k>=(1LL<<Bit))Bit++;
    Bit--;
    dp[0][0]=1;
    while(~Bit){
        if((k>>Bit)&1)
        Add();
        if(Bit)
        Mul();
        Bit--;
    }
    _for(i,0,n){
        _for(j,0,n)
        ans[(i+j)%n]=(ans[(i+j)%n]+1LL*x[i]*dp[pos][j])%mod;
    }
    int inv=quick_pow(quick_pow(a+b+c,mod-2),k);
    _for(i,0,n)space(1LL*ans[i]*inv%mod);
    return 0;
}
```

### 题解 2

循环矩阵快速幂，时间复杂度 $O(n^2\log k)$

$$ \begin{pmatrix} \text{ans}_0\\ \text{ans}_1\\ \vdots\\ \text{ans}_{n-1} \end{pmatrix}= \begin{pmatrix} c & b & 0 & \cdots & a \\ a & c & b & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b & 0 & 0 & \cdots & c \end{pmatrix} \begin{pmatrix} x_0\\ x_1\\ \vdots\\ x_{n-1} \end{pmatrix}$$

```
const int MAXN=505,mod=998244353;
int quick_pow(int a,LL b){
    int ans=1;
    while(b){
        if(b&1)ans=1LL*ans*a%mod;
        a=1LL*a*a%mod;
        b>>=1;
    }
    return ans;
}
int n,x[MAXN],y[MAXN],z[MAXN],ans[MAXN],a,b,c,pos;
```

```
void Mul(int *a,int *b){
    static int temp[MAXN];
    mem(temp,0);
    _for(i,0,n){
        _for(j,0,n)
        temp[i]=(temp[i]+1LL*a[j]*b[(i-j+n)%n])%mod;
    }
    _for(i,0,n)a[i]=temp[i];
}
int main()
{
    n=read_int();
    LL k=read_LL(),t=k;
    a=read_int(),b=read_int(),c=read_int();
    y[0]=1,z[0]=c,z[n-1]=a,z[1]=b;
    _for(i,0,n)x[i]=read_int();
    while(t){
        if(t&1)Mul(y,z);
        Mul(z,z);
        t>>=1;
    }
    _for(i,0,n){
        _for(j,0,n)
        ans[i]=(ans[i]+1LL*x[j]*y[(j-i+n)%n])%mod;
    }
    int inv=quick_pow(quick_pow(a+b+c,mod-2),k);
    _for(i,0,n)space(1LL*ans[i]*inv%mod);
    return 0;
}
```

## 题解 3

设 $F=\{x_0,x_1\cdots x_{n-1}\},G=\{c,a,0\cdots 0,b\}$

于是所求答案为 $FG^k$ 的长度为 $n$ 的循环卷积。

直接暴力卷积时间复杂度 $O(n\log n\log k)$如果套用 $\text{Bluestein's Algotithm}$ 时间复杂度 $O(n(\log n+\log k))$