Atcoder Rugular Contest 113

比赛链接

F - Social Distance

题意

给定 $n+1$ 个数 $X_0,X_1\cdots X_n(X_0=0)$，$y_i\in [X_{i-1},X_i]$ 且 y_i 取值随机，求

$$ E\left(\min_{2\le i\le n}(y_i-y_{i-1})\right)\bmod 998244353 $$

题解

设 $F(x)=P\{\min_{2\le i\le n}(y_i-y_{i-1})\gt x\}$，于是有

$$ E\left(\min_{2\le i\le n}(y_i-y_{i-1})\right)=\int-xF'(x)\mathrm{d}x $$

问题转化为求 $F(x)$ 的表达式。不妨先考虑固定 x，计算 $F(x)$ 的值。

记 $z_i=y_i-(i-1)x$，于是 $\min_{2\le i\le n}(y_i-y_{i-1})\gt x$ 等价于 $z_1\lt z_2\cdots \lt z_n,z_i\in [X_{i-1}-(i-1)x,X_i-(i-1)x]$，

将所有 $[X_{i-1}-(i-1)x,X_i-(i-1)x]$ 的左右端点 t_{2i-1},t_{2i} 进行排序，得到 $v_1,v_2\cdots v_{2n}$，同时不妨假设 $z_i\neq v_j$，这样不影响概率。

设 $\text{dp}(i,j,k)$ 表示 $v_j\lt z_{i-k+1}\lt z_{i-k+2}\cdots \lt z_i\lt v_{j+1}$ 的概率。

于是若 $v_j\lt z_{i-k+1}\lt z_{i-k+2}\cdots \lt z_i\lt z_{i+1}\lt v_{j+1}$，则有

$$ \text{dp}(i+1,j,k+1)\gets [X_i-ix\le v_j\lt X_{i+1}-ix]\frac {(v_{j+1}-v_j)}{(k+1)(X_{i+1}-X_i)}\text{dp}(i,j,k) $$

若 $v_j\lt z_{i-k+1}\lt z_{i-k+2}\cdots \lt z_i\lt v_{j+1}\cdots\lt v_{j+t}\lt z_{i+1}\lt v_{j+t+1}$，则有

$$ \text{dp}(i+1,j+t,1)\gets [X_i-ix\le v_{j+t}\lt X_{i+1}-ix]\frac {v_{j+t+1}-v_{j+t}}{X_{i+1}-X_i}\text{dp}(i,j,k) $$

利用前缀和可以 $O(n^3)$ 完成转移。代码实现中优化了 dp 第三维，与上述转移略有不同，但时间复杂度也是 $O(n^3)$。

接下来考虑 x 不固定的情况，由于 v_j 总是 $b-kx$ 的形式，于是 $v_{j+1}-v_j$ 也是 x 的一次项。

dp 转移中每次转移一定有 $i\to i+1$，而每次转移中含有 x 项只有 $v_{j+1}-v_j$，于是 $F(x)$ 最多是 x 的 n 次多项式。

当所有 $t_i(x)$ 的相对大小关系不变时，易知 $F(x)$ 的表示式是固定的。于是不妨两两枚举使得 $t_i(x)=t_j(x)$ 的情况，此时有 $x=\frac {b_i-b_j}{k_j-k_i}$。

保存所有上述 x 中非负的数，排序后去重，记为 $x_0,x_1\cdots x_k(x_0=0)$于是有 $k\sim O(n^2)$

对于 $x\in [x_i,x_{i+1}]$先令 $x=\frac {x_i+x_{i+1}}2$然后计算出 $t_i(x)$ 的值后排序，得到 t_i 的名次。

然后考虑计算 $F(0\sim n)$注意此时 dp 转移判定 $[X_i-ix\le v_j\lt X_{i+1}-ix]$ 不能靠权值要靠名次，因为取模意义下权值没有意义。

拉格朗日插值法计算 $F(x)$ 在 $x\in [x_i,x_{i+1}]$ 的表达式，最后计入贡献

$$ \text{ans}\gets \int_{x_i}^{x_{i+1}}-xF'(x)\mathrm{d}x $$

总时间复杂度 $O(n^6)$

```
const int MAXN=25,MAXV=1e6+5,Mod=998244353;
int inv[MAXV];
int quick_pow(int a,int k){
    int ans=1;
    while(k){
        if(k&1)ans=1LL*ans*a%Mod;
        a=1LL*a*a%Mod;
        k>>=1;
    }
    return ans;
}
int gcd(int a,int b){
    while(b){
        int t=b;
        b=a%b;
        a=t;
    }
    return a;
}
struct Poly{
    int a[MAXN];
    Poly(){mem(a,0);}
    Poly operator * (const Poly &b)const{
        Poly c;
        _for(i,0,MAXN)_rep(j,0,i)
        c.a[i]=(c.a[i]+1LL*a[j]*b.a[i-j])%Mod;
        return c;
    }
    int cal(int x){
        int ans=0;
        for(int i=MAXN-1;i>=0;i--)
        ans=(1LL*ans*x+a[i])%Mod;
        return ans;
    }
}p[MAXN],F;
struct Frac{
```

```
    int p,q;
    Frac(int p=0,int q=1){
        int g=gcd(p,q);
        p/=g,q/=g;
        if(q<0)p=-p,q=-q;
        this->p=p;
        this->q=q;
    }
    bool operator < (const Frac &b)const{
        return 1LL*p*b.q<1LL*b.p*q;
    }
    bool operator == (const Frac &b)const{
        return p==b.p&&q==b.q;
    }
};
struct Node{
    double v;int idx;
    bool operator < (const Node &b)const{
        return v<b.v;
    }
}node[MAXN<<1];
vector<Frac> vec;
vector<pair<int,int> >b;
int a[MAXN],val[MAXN<<1],rk_l[MAXN],rk_r[MAXN];
int y[MAXN],dp[MAXN][MAXN];
int solve(int n){
    mem(dp,0);
    dp[0][0]=1;
    _for(j,0,(n<<1)-1){
        _rep(i,0,n){
            int g=1;
            dp[i][j+1]=(dp[i][j+1]+dp[i][j])%Mod;
            _rep(k,1,n-i){
                if(rk_l[i+k]>j||rk_r[i+k]<j+1)break;
                g=1LL*g*inv[k]%Mod*(val[j+1]-val[j])%Mod;
                dp[i+k][j+1]=(dp[i+k][j+1]+1LL*dp[i][j]*g)%Mod;
            }
        }
    }
    return dp[n][(n<<1)-1];
}
int cal(int n,int v1,int v2){
    mem(F.a,0);
    _rep(i,0,n){
        _rep(j,0,n)
        F.a[i]=(F.a[i]+1LL*y[j]*p[j].a[i])%Mod;
    }
    for(int i=MAXN-2;i;i--)
    F.a[i+1]=1LL*F.a[i]*i%Mod*inv[i+1]%Mod;
    F.a[1]=F.a[0]=0;
    return (Mod+F.cal(v2)-F.cal(v1))%Mod;
```

```
}
int main()
{
    int n=read_int(),ans=0;
    inv[1]=1;
    _for(i,2,MAXV)inv[i]=1LL*(Mod-Mod/i)*inv[Mod%i]%Mod;
    _rep(i,0,n){
        p[i].a[0]=1;
        _rep(j,0,n){
            if(i==j)continue;
            Poly temp;
            temp.a[0]=1LL*(Mod-j)*quick_pow((Mod+i-j)%Mod,Mod-2)%Mod;
            temp.a[1]=quick_pow((Mod+i-j)%Mod,Mod-2);
            p[i]=p[i]*temp;
        }
    }
    _rep(i,0,n)a[i]=read_int();
    _rep(i,1,n){
        b.push_back(make_pair(a[i-1],-(i-1)));
        b.push_back(make_pair(a[i],-(i-1)));
    }
    _for(i,0,b.size()){
        _for(j,i+1,b.size()){
            pair<int,int> p1=b[i],p2=b[j];
            if(p1.second!=p2.second)
            vec.push_back(Frac(p1.first-p2.first,p2.second-p1.second));
        }
    }
    vec.push_back(Frac(0,1));
    sort(vec.begin(),vec.end());
    vec.erase(unique(vec.begin(),vec.end()),vec.end());
    _for(i,0,vec.size()){
        if(vec[i].p<=0)continue;
        double
x=(1.0*vec[i-1].p*vec[i].q+1.0*vec[i-1].q*vec[i].p)/(2.0*vec[i-1].q*vec[i].
q);
        _for(j,0,b.size())
        node[j].v=b[j].first+x*b[j].second,node[j].idx=j;
        sort(node,node+b.size());
        _for(j,0,b.size()){
            if(node[j].idx&1)
            rk_r[node[j].idx/2+1]=j;
            else
            rk_l[node[j].idx/2+1]=j;
        }
        _rep(j,0,n){
            _rep(k,1,n){
                val[rk_l[k]]=(a[k-1]+1LL*(Mod-(k-1))*j)%Mod;
                val[rk_r[k]]=(a[k]+1LL*(Mod-(k-1)*j))%Mod;
            }
```

```
            y[j]=solve(n);
        }
ans=(ans+cal(n,1LL*vec[i-1].p*inv[vec[i-1].q]%Mod,1LL*vec[i].p*inv[vec[i].q
]%Mod))%Mod;
    }
    _rep(i,1,n)
    ans=1LL*ans*inv[a[i]-a[i-1]]%Mod;
    enter((Mod-ans)%Mod);
    return 0;
}
```

From:
https://wiki.cvbbacm.com/ - **CVBB ACM Team**

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:jxm2001:contest:arc_113

Last update: **2021/02/24 10:36**