2020.06.08-2020.06.14 周报

团队训练

2020.06.14 2017-2018 ACM-ICPC, NEERC, Moscow Subregional Contest prob:6:6:10 rnk:124/?

20200614比赛记录

_wzx27

1.指数生成函数

常用于解决一类多重集合计数的问题

设 S 是多重集合$\{n_1\cdot a_1,n_2\cdot a_2,\cdots,n_k\cdot a_k\}$其中 $n_1,n_2,\cdots,n_k$ 是非负整数。设 h_n 是 S 的 n 排列数。那么数列 $h_0,h_1,\cdots,h_n,\cdots$ 的指数生成函数 $g^{(e)}(x)$ 由下式给出 $$g^{(e)}(x)=f_{u_1}(x)f_{u_2}(x)\cdots f_{u_k}(x)$$ 其中，对于 $i=1,2,\cdots,k$有 $$f_{u_i}(x)=1+x+\frac {x^2}{2!}+\cdots + \frac {x^{n_i}}{n_i!}$$

又因为 $e^x=1+x+\frac {x^2}{2!}+\cdots$所以这类因子相乘有很好的性质，便于求解最后的生成函数。

POJ3734

用红、蓝、绿、黄四种颜色给 $1\times n$ 的方块染色，红色和绿色方块的数目必须是偶数。

那么蓝黄的因子为 $1+x+\frac {x^2}{2!}+\cdots + \frac {x^{n}}{n!}+\cdots=e^x$ 红绿的因子为 $1+\frac {x^2}{2!} + \frac{x^4}{4!} \cdots=\frac {e^x+e^{-x}}2$ 所以生成函数为 $g^{(e)}(x)=e^{2x}(\frac {e^x+e^{-x}}2)^2=\frac {e^{4x}+2e^{2x}+2}4$ 从而展开后 $n\gt 1$ 时 $\frac{x^n}{n!}$ 的系数为 $\frac {4^n+2^{n+1}}4$

POJ1322

c 种无限集合，任意选取 n个，相同类型的元素出现两个时就会一起消失，问最后剩下 m 个的概率是多少。

范围:$c\le 100,n,m\le 1e6$

可以概率dp但好像很麻烦？

考虑生成函数的做法，就是有 m 个集合取了奇数次，另外 $c-m$ 个集合取了偶数次。

$$ \begin{aligned} g^{(e)}(x)=&(\frac {e^x-e^{-x}}2)^m\cdot (\frac{e^x+e^{-x}}2)^{c-m}\\ =&2^{-c}\sum_{i=0}^{m}(-1)^{m-i}{m \choose i}e^{ix}e^{-(m-i)x}\cdot \sum_{j=0}^{c-m}{c-m \choose j}e^{jx}e^{-(c-m-j)x} \end{aligned} $$

然后暴力 $O(c^2logn)$ 就可以求每一对 (i,j) 对 $\frac {x^n}{n!}$ 的系数的贡献 $(-1)^{m-i}{m \choose i}{c-m \choose j}(2i+2j-c)^n$

```
#include<iostream>
#include<algorithm>
#include<iomanip>
#define ll long long
#define pii_ pair<int,int>
#define mp_ make_pair
#define pb push_back
#define fi first
#define se second
#define rep(i,a,b) for(int i=(a);i<=(b);i++)
#define per(i,a,b) for(int i=(a);i>=(b);i--)
#define show1(a) cout<<#a<<" = "<<a<<endl
#define show2(a,b) cout<<#a<<" = "<<a<<"; "<<#b<<" = "<<b<<endl
using namespace std;
const ll INF = 1LL<<60;
const int inf = 1<<30;
const int maxn = 2e5+5;
inline void fastio() {ios::sync_with_stdio(false);cin.tie(0);cout.tie(0);}

double qpow(double a,ll b) {double
s=1;while(b){if(b&1)s=s*a;a=a*a;b>>=1;}return s; }
int c,n,m;
double C[105][105];
int main()
{
    fastio();
    rep(i,0,100) C[i][0]=1;
    rep(i,1,100) rep(j,1,i) C[i][j] = C[i-1][j-1]+C[i-1][j];
    while(cin>>c && c){
        cin>>n>>m;
        if(m>c||m>n||(n-m)&1) {cout<<"0.000"<<endl;continue;}
        double ans = 0.0;
        rep(i,0,c-m){
            rep(j,0,m){
                double t = 1.0*(2*i+2*j-c)/c;
                if((m-j)&1) ans -= qpow(t,n)*C[c-m][i]*C[m][j];
                else ans += qpow(t,n)*C[c-m][i]*C[m][j];
            }
        }
        ans *= 1.0*C[c][m]/qpow(2,c);
        cout<<setprecision(3)<<ans<<endl;
    }
    return 0;
}
```

2.生成函数与多项式

生成函数没那么好求的时候就要用多项式全家桶

P4389付公主的背包

n 种无限个物品，每个物品体积为 v_i求恰好装下体积为 $1~m$ 的种数。

数据范围$n,m \le 1e5$

这种题也算是生成函数的经典应用

首先得到生成函数$f(x)=\prod _{i=1}^n \frac 1{1-x^{v_i}}$

两边取对数得 $$ \begin{aligned} lnf(x) =&\sum_{i=1}^n\sum_{j=0}^\infty \frac {x^{j\cdot v_i}}j \\ =&\sum_{k=0}^m \sum _{v_i|k} \frac {v_i}k x^k \pmod{x^{m+1}} \end{aligned} $$

$O(nlogn)$预处理出 k 的所有因子，然后多项式 $\text{Exp} $求 $e^{lnf(x)}$ 即可。

```
#include<bits/stdc++.h>
#define ll long long
#define pii_ pair<int,int>
#define mp_ make_pair
#define pb push_back
#define fi first
#define se second
#define rep(i,a,b) for(int i=(a);i<=(b);i++)
#define per(i,a,b) for(int i=(a);i>=(b);i--)
#define show1(a) cout<<#a<<" = "<<a<<endl
#define show2(a,b) cout<<#a<<" = "<<a<<"; "<<#b<<" = "<<b<<endl
using namespace std;
const ll INF = 1LL<<60;
const int inf = 1<<30;
const int maxn = 4e5+5;
const ll M = 998244353;
inline void fastio() {ios::sync_with_stdio(false);cin.tie(0);cout.tie(0);}
ll qpow(ll a,ll b) {ll
s=1;while(b){if(b&1)s=(s*a)%M;a=(a*a)%M;b>>=1;}return s; }
int n,m,cnt[maxn],r[maxn];
ll A[maxn],B[maxn],tmp[maxn],t1[maxn],t2[maxn],t3[maxn];
vector<int> g[maxn];
void init()
{
    int n = 1e5;
    rep(i,1,n){
        for(int j=i;j<=n;j+=i) g[j].pb(i);
    }
}
void add(ll &x,ll y)
{
    x+=y;if(x<0) x+=M;if(x>M) x-=M;
}
void NTT(ll a[],int lim,int type)
```

```
{
    rep(i,0,lim-1) if(i<r[i]) swap(a[i],a[r[i]]);
    for(int mid=1;mid<lim;mid<<=1){
        ll wn = qpow(3,(M-1)/mid/2);
        if(type==-1) wn = qpow(wn,M-2);
        for(int R=mid<<1,j=0;j<lim;j+=R){
            ll w = 1;
            for(int k=0;k<mid;k++,w=w*wn%M){
                ll x=a[j+k],y=w*a[j+mid+k]%M;
                a[j+k] = (x+y)%M;
                a[j+mid+k] = (x-y+M)%M;
            }
        }
    }
    if(type==-1){
        ll inv = qpow(lim,M-2);
        rep(i,0,lim) a[i]=a[i]*inv%M;
    }
}
void PolyMul(ll a[],ll b[],int n,int m)
{
    int lim=1,l=0;
    while(lim<=n+m) lim<<=1,l++;
    rep(i,1,lim) r[i] = (r[i>>1]>>1) | ((i&1)<<(l-1));
    NTT(a,lim,1);NTT(b,lim,1);
    rep(i,0,lim) a[i] = a[i]*b[i]%M;
    NTT(a,lim,-1);
}
void PolyInv(int n,ll a[],ll b[])
{
    if(n==1) {b[0]=qpow(a[0],M-2);return ;}
    PolyInv((n+1)>>1,a,b);
    int lim=1,l=0;
    while(lim<2*n) lim<<=1,l++;
    rep(i,1,lim-1) r[i] = (r[i>>1]>>1) | ((i&1)<<(l-1));
    rep(i,0,n-1) tmp[i] = a[i];
    rep(i,n,lim) tmp[i] = 0;
    NTT(tmp,lim,1);NTT(b,lim,1);
    rep(i,0,lim){
        b[i] = (2-tmp[i]*b[i]%M+M)%M*b[i]%M;
    }
    NTT(b,lim,-1);
    rep(i,n,lim) b[i]=0;
}
void PolyDf(int n,ll a[],ll b[])
{
    rep(i,0,n) b[i] = a[i+1]*(i+1)%M;
}
void PolyItg(int n,ll a[],ll b[])
```

```
{
    rep(i,1,n) b[i] = a[i-1]*qpow(i,M-2)%M;b[0]=0;
}
void PolyLn(int n,ll a[],ll b[])
{
    PolyDf(n,a,t1);
    PolyInv(n,a,t2);
    PolyMul(t1,t2,n,n);
    PolyItg(n,t1,b);
}
void PolyExp(int n,ll a[],ll b[])
{
    if(n==1) {b[0]=1;return ;}
    PolyExp((n+1)>>1,a,b);
    int lim=1,l=0;
    while(lim<2*n) lim<<=1,l++;
    rep(i,0,lim) t1[i]=t2[i]=0;
    PolyLn(n,b,t3);
    rep(i,1,lim) r[i] = (r[i>>1]>>1) | ((i&1)<<(l-1));
    rep(i,1,n-1) t3[i] = (a[i]-t3[i]+M)%M;
    t3[0] = (1+a[0]-t3[0]+M)%M;
    rep(i,n,lim) b[i]=t3[i]=0;
    NTT(t3,lim,1);NTT(b,lim,1);
    rep(i,0,lim) b[i] = b[i]*t3[i]%M;
    NTT(b,lim,-1);
    rep(i,n,lim) b[i] = 0;
}
int main()
{
    fastio();
    init();
    cin>>n>>m;
    int x;
    rep(i,1,n) {cin>>x;cnt[x]++;}
    rep(i,0,m){
        ll inv = qpow(i,M-2);
        for(int x:g[i]) if(cnt[x]) add(A[i],cnt[x]*x%M*inv%M);
    }
    PolyExp(m+1,A,B);
    rep(i,1,m) cout<<B[i]<<endl;
    return 0;
}
```

Infinity37

D.Decoding of Varints

Zars19

C.Carpet

本周推荐