组合数学相关

相关定理

定理1

$\{1,2,...n\}$的r组合$a_1,a_2,...,a_r$出现在所有r组合中的字典序位置编号：
$index = C(n,r) - C(n-a_1,r) - C(n-a_2,r-1) - ... - C(n-a_r,1)$

定理2

$k*C(n,k) = n*C(n-1,k-1)$
$C(n,0)+C(n,2)+... = C(n,1)+C(n,3)+...$
$1*C(n,1)+2*C(n,2)+...+n*C(n,n) = n*2^{n-1}$

错位排列

$D_n = n!\sum_{k=0}^{n}\frac{(-1)^k}{k!}$

$D_n = (n-1)*(D_{n-1}+D_{n-2})$

Catalan数

凸多边形三角划分,n个节点组成二叉搜索树，n对括号正确匹配数目，1-n的出栈序列
$C_n = \frac{C(2*n,n)}{n+1}$
$C_n = \frac{(4*n-2)}{n+1}*C_{n-1}$
$C_1 = 1$

Stirling数

第一类Stirling数

s(p, k)将 p 个不同元素构成 k 个圆排列的数目。
$s(p,k) = (p-1)*s(p-1,k) + s(p-1,k-1)$

第二类Stirling数

n个元素拆分k个集合的方案数记为S(n, k)。
$S(p,k) = k*S(p-1,k) + S(p-1,k-1)$
$S(p,0) = 0, (p >= 1)$ $S(p,p) = 1, (p >= 0)$
$S(p,1) = 1, (p >= 1)$ $S(p,2) = 2^{p-1}-1, (p >= 2)$
$S(p,p-1) = C(p,2)$

Bell数

元素个数为n的集合的划分数目。 $B_p = \sum_{i=0}^{p} S(p,i)$
其中$S(p,i)$表示第二类Stirling数。
$B_p = C(p-1,0)*B_0 + C(p-1,1)*B_1 + ... + C(p-1,p-1)*B_{p-1}$

卢卡斯定理

若p是质数：$C(n,m)\%p=C(n/p,m/p)*C(n\%p,m\%p)\%p$

若p不是质数，令$p=p_1*p_2*...*p_n$，这里p_i是质数$\begin{cases} C_n^m \equiv a_1 \pmod{p_1} \\ C_n^m \equiv a_2 \pmod{p_2} \\ \quad\vdots \\ C_n^m \equiv a_n \pmod{p_n} \end{cases}$则分别求出$a_1,a_2,...,a_n$用中国剩余定理合并即可。

3.1. 盒子放小球问题

n个小球，m个盒子。

3.1.1 n个小球有区别，m个盒子有区别

(1)允许空盒：每个球放到任意盒子里，总方案数m^n。
(2)不允许空盒：需满足$n \geq m \geq 1$，$m>n$时无解。其方案数及时看成m个盒子相同时的方案数，再乘以$m!$。答案即是$S(n,m)*m!$。S代表第二类斯特林数。

3.1.2 n个小球有区别，m个盒子无区别

(1)允许空盒：假设放了k个盒，$m \geq k \geq 1$。那么答案就是$\sum_{k=1}^{m} S(n,k)$。
(2)不允许有空盒：$S(n,m)$。

3.1.3 n个小球无区别，m个盒子有区别

(1)允许空盒：$n \geq m \geq 1$。“隔板法”。假设不允许有空盒，每一个盒里都先放一个小球，这样小球共有$n+m$个，然后插板，插板的方案数为C_{n+m-1}^{n-1}。
(2)不允许空盒：$n \geq m \geq 1$。“隔板法”。方案数C_{n-1}^{m-1}。

3.1.4 n个小球无区别，m个盒子无区别

(1)允许空盒：划分数问题。$dp[i][j]$表示i个球，j个盒子的方案数。转移方程为
$dp[i][j]=dp[i-j][j]+dp[i][j-1](i \geq j)$
$dp[i][j]=dp[i][j-1](i<j)$
如果$n<m$，答案为$dp[n][n]$，否则为$dp[n][m]$。
(2)不允许空盒：$n \geq m \geq 1$。转成上情况的$n-m$个小球，m个盒子。

3.2. 计数原理与计数公式

3.2.1 可重复的排列与组合

3.2.1.1 可重复的排列

从n个不同元素中取m个元素（同一元素可以重复取出），按照一定的顺序排成一列。排列的个数为n^m。

3.2.1.2 可重复的组合

从n个不同元素中取m个元素（同一元素可以重复取出），并成一组。组合的个数为C_{n+m-1}^{m}。
【证明】
$1,2,...,n$表示n个不同元素。从中取m个可以表示成：
$\{i_1,i_2,...,i_m\}(1 \leq i_1 \leq i_2 \leq ... \leq i_m \leq n)$
令$j_k=i_k+(k-1)$，即：

$j_1 = i_1$
$j_2 = i_2 + 1$
$j_3 = i_3 + 2$
$\ldots$
$j_m = i_m + (m-1)$
可以得到组合
$\{j_1, j_2, \ldots, j_m\}(1 \leq j_1 < j_2 < \ldots < j_m \leq n-m+1)$
这样就相当于在$n+m-1$个元素中取m个不相同的元素，作为一组。
因此即是C_{n+m-1}^{m}。

3.2.1.3 不全相异元素的全排列

n个元素中，分别有$n_1, n_2, \ldots, n_k$个元素相同，且$n_1 + n_2 + \ldots + n_k = n$，则称这n个元素的全排列为不全相异元素的全排列，个数为
$\frac{n!}{n_1!n_2!\ldots n_k!}$

3.2.1.4 多组组合

n个相异的元素分为$k(k \leq n)$个按照**一定顺序**排列的组，其中第i组有n_i个元素
$(i = 1, 2, \ldots, k)(n_1 + n_2 + \ldots + n_k = n)$。不同的分组方法为
$\frac{n!}{n_1!n_2!\ldots n_k!}$
【例】
从$n(n \geq 6)$个选手中选3对选手参加双打，问共有多少种选法。
答案为（注意不考虑**组的顺序**）
$\frac{C_n^6 * \frac{6!}{2!*2!*2!}}{3!}$

3.2.2 相异元素的圆排列和项链数

3.2.2.1 圆排列

n个元素不分首尾排成一圈，成为n个相异元素的圆排列。排列的种数为$(n-1)!$。

3.2.2.2 项链数

将n粒不相同的珠子，穿成一副项链，得到的不同的项链数。
由于项链顺时针和逆时针都是相同的，所以个数即是圆排列的一半。
$$\begin{cases} 1, n = 1 \text{或} n = 2 \\ \dfrac{1}{2} * (n-1)!, n \geq 3 \end{cases}$$

3.2.3 错排问题

错排递推式。
$D(n)$代表n个数的错排公式，则
$D(n) = (n-1) * [D(n-1) + D(n-2)]$
错排公式
$D(n) = n!(1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \ldots + \frac{(-1)^n}{n!})$

3.2.4 组合数常用公式

$$C_n^2=\frac{n*(n-1)}{2}$$

$$C_n^3=\frac{n*(n-1)(n-2)}{6}$$

$$C_n^m=C_{n-1}^{m-1}+C_{n-1}^m$$

$$m*C_n^m=n*C_{n-1}^{m-1}$$

$$C_n^0+C_n^1+...+C_n^n=2^n$$

$$1C_n^1+2C_n^2+...+nC_n^n=n2^{n-1}$$

$$1^2C_n^1+2^2C_n^2+...+n^2C_n^n=n(n+1)2^{n-2}$$

$$\frac{C_n^1}{1}-\frac{C_n^2}{2}+\frac{C_n^3}{3}+...+(-1)^{n-1}\frac{C_n^n}{n}=1+\frac{1}{2}+\frac{1}{3}+...+\frac{1}{n}$$

$$(C_n^0)^2+(C_n^1)^2+(C_n^2)^2+...+(C_n^n)^2=C_{2n}^n$$

范德蒙恒等式:
$\sum_{i=0}^k C_n^i C_m^{k-i}=C_{n+m}^k$

经验式(link https://www.cnblogs.com/qrsikno/p/10170523.html):

$$\sum_{i=0}^n C_n^i*r^i=(r+1)^n\text{（广义二项式定理）}$$

$$\sum_{i=0}^n i*C_n^i=n*2^{n-1}$$

$$\sum_{i=0}^n C_i^k=C_{n+1}^{k+1}$$

$$\sum_{i=0}^k C_{n+i}^i=C_{n+k+1}^k$$

3.3. 抽屉原理与平均值原理

3.3.1 抽屉原理

3.3.1.1 第一抽屉原理

如果将m个物件放入n个抽屉内，那么必有一个抽屉内至少有$[\frac{m-1}{n}]+1$个物件。
【推广】
如果将$m_1+m_2+...+m_n+1$个物件放入n个抽屉内，那么或者第一个抽屉内至少有m_1+1个物件，或者第二个抽屉内至少有m_2+1个物件……或者第n个抽屉内至少有m_n+1个物件。

3.3.1.2 第二抽屉原理

如果将m个物件放入n个抽屉内，那么必有一个抽屉内至多有$[\frac{m}{n}]$个物件。
【推广】
如果将$m_1+m_2+...+m_n-1$个物件放入n个抽屉内，那么或者第一个抽屉内至多有m_1-1个物件，或者第二个抽屉内至多有m_2-1个物件……或者第n个抽屉内至多有m_n-1个物件。

3.3.2 平均值原理

（1）设$a_1,a_2,...,a_n$是实数，$A=\frac{1}{n}(a_1+a_2+...+a_n)$，则$a_1,a_2,...,a_n$中必有一个数不小于A，也有一个数不大于A。
（2）设$a_1,a_2,...,a_n$是实数，$G=\frac{1}{n}\sqrt[n]{a_1a_2...a_n}$，则$a_1,a_2,...,a_n$中必有一个数不小于G，也有一个数不大于G。

3.4. 生成函数

生成函数的定义：
实数序列 $a_0,a_1,...,a_k,...$的生成函数是无穷级数
$G(x)=a_0+a_2x+...+a_kx^k+...=\sum_{k=0}^{\infty}a_kx^k$
a_k的普通生成函数。
广义二项式系数：

$$\binom{u}{k}=\begin{cases}u(u-1)(u-2)...(u-k+1)/k!, & k>0\\ 1, & k=0\end{cases}$$

【例】

$$\begin{aligned}\binom{1/2}{3}&=\frac{(1/2)(1/2-1)(1/2-2)}{3!}\\&=\frac{(1/2)(-1/2)(-3/2)}{6}\\&=1/16\end{aligned}$$

设x是实数，$|x|<1$，u是实数，那么
$(1+x)^u=\sum_{k=0}^{\infty}\binom{u}{k}x^k$

3.4.1 常用生成函数

$$\frac{1-x^{n+1}}{1-x}=\sum_{k=0}^{n}x^k$$

$$\frac{1}{1-ax}=\sum_{k=0}^{\infty}a^kx^k$$

$$\frac{1}{(1-x)^2}=\sum_{k=0}^{\infty}(k+1)x^k$$

$$\frac{1}{(1-x)^n}=\sum_{k=0}^{\infty}C_{n+k-1}^{k}x^k$$

$$\frac{1}{(1+x)^n}=\sum_{k=0}^{\infty}C_{n+k-1}^{k}(-1)^kx^k$$

3.4.2 计数问题

3.5. 特殊计数序列

3.5.1 Catalan数列

前几项：$1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, 16796, 58786, 208012, \ldots\ldots$即 $c[0] = 1, c[1] = 1, c[2] = 2\ldots$

递推式1：$f[n] = \sum_{i=0}^{n-1} f[i] * f[n-i-1]$

递推式2：$f[n] = \frac{4n-2}{n+1} f[n-1]$

组合式1：$f[n] = \frac{C_{2n}^{n}}{n+1}$

组合式2：$f[n] = C_{2n}^{n} - C_{2n}^{n-1}$

应用：

1. 二叉树计数1：已知二叉树有n个节点，能够构成C_n种不同的二叉树。（二叉搜索树）
2. 二叉树计数2：已知二叉树的叶子n个，能够构成C_{n-1}种不同的二叉树。（二叉搜索树）
3. 括号匹配数：一个合法的表达式由()包围，()可以嵌套和连接，给出n对括号，可以组成的合法表达式的个数为C_n。
4. 划分问题：将一个凸$n+2$多边形区域分成三角形区域的方法数为C_n。
5. 出栈问题1：一个栈的进栈序列为$1, 2, 3, ..n$，不同的出栈序列有C_n种。
6. 出栈问题2：有2n个人排成一行进入剧场。入场费5元。其中只有n个人有一张5元钞票，另外n人只有10元钞票，剧院无其它钞票，问有多少种方法使得只要有10元的人买票，售票处就有5元的钞票找零。5元的相当于入栈，10元的相当于出栈，转化成上问题。
7. 路径问题：在$n*n$的方格地图中，从一个角到另外一个角，不跨越对角线的路径数有C_n种。
8. 握手问题：$2n$个人均匀坐在一个圆桌边上，某个时刻所有人同时与另一个人握手，要求手之间不能交叉，共有C_n种握手方法。

3.5.2 Fibonacci数列

通项公式：$F_n = \frac{1}{\sqrt{5}}[(\frac{1+\sqrt{5}}{2})^n - (\frac{1-\sqrt{5}}{2})^n]$
递推式：
$F_n = F_{n-1} + F_{n-2}$
性质：
$F_1 + F_1 + F_2 + F_3 + \ldots + F_n = F_{n+2} - 1$

$F_1 + 2F_2 + 3F_3 + \ldots + nF_n = nF_{n+2} - F_{n+3} + 2$

$F_1 + F_3 + F_5 + \ldots + F_{2n-1} = F_{2n}$

$F_2 + F_4 + F_6 + \ldots + F_{2n} = F_{2n+1} - 1$

$F_1^2 + F_2^2 + F_3^2 + \ldots + F_n^2 = F_n F_{n+1}$

$F_{n-1} F_{n+1} - F_n^2 = (-1)^n$

定理：
$F_nF_m + F_{m-1}F_{n-1} = F_{m+n-1}$

$F_mF_{n+1} + F_{m-1}F_n = F_{m+n}$

$m = n$时，
$F_{2n-1} = F_n^2 + F_{n-1}^2$

$F_{2n} = (F_{n-1} + F_{n+1})F_n = (2F_{n-1} + F_n)F_n$

F_n整除F_m当且仅当n整除m，其中$n \geq 3$

任意连续三个$Fibonacci$数两两互素。

3.5.3 Lucas数列

定义：
$L_n = \begin{cases} 2, & n = 1 \\ 1, & n = 2 \\ L_{n-1} + L_{n-2}, & n \geq 3 \end{cases}$
通项公式：
$L_n = (\frac{1+\sqrt{5}}{2})^n + (\frac{1-\sqrt{5}}{2})^n$
与Fibonacci数的关系：
$F_{2n} = L_nF_n$

$L_n = F_{n-1} + F_{n+1}$

$F_n = \dfrac{L_{n-1} + L_{n+1}}{5}$

$L_n^2 = 5F_n^2 + 4(-1)^n$

3.5.4 Stirling数

3.5.4.1 第一类Stirling数

$S1(n, m)$表示的是将n个不同元素构成m个圆排列的数目。
递推式：
$S1(n, m) = (n-1) * S1(n-1, m) + S1(n-1, m-1)(n > 1, m > 1)$

边界条件：
$S1(0, 0) = 1, S1(n, 0) = 0$
$S1(n, n) = 1$
性质：
$\sum_{k=0}^{n} S1(n, k) = n!$
【例】n个仓库， $2n$把钥匙，n 位官员。如果把n位官员分成m个不同的部，部中的官员数量与管理的仓库数量一致。有多少种方案使得所有同部的官员可以打开所有本部管理的仓库，而无法打开其他管理的仓库。（n把钥匙放到仓库，n把钥匙分给官员）
方案数即为$S1(n, m)n!$。
前面的是放到仓库里的方案数，后面说官员的分配方案。

3.5.4.2 第二类Stirling数

$S2(n,m)$表示的是把n个不同元素划分到m个集合的方案数。
递推式：
$S2(n,m)=m*S2(n-1,m)+S2(n-1,m-1)(1\le m\le n-1)$
边界条件：
$S2(n,0)=0,S2(n,1)=1$
$S2(n,n)=1$

欧拉相关

欧拉函数

1.相关公式

$\varphi(x)=x\prod_{i=0}^{n}(1-\frac{1}{p_i})$

$\varphi(2n)=\varphi(n)$

若 p 是质数，$\varphi(p^k)=p^k-p^{k-1}$

小于 n 的数中，与 n 互质的数的总和为 $\frac{n\varphi(n)}{2}$

欧拉反演：$n=\sum_{d|n}\varphi(d)$

若 a,p 互质，$a^{\varphi(p)}\equiv 1\pmod p$，即 $a^b\equiv a^{b \bmod \varphi(p)}\pmod p$

扩展欧拉定理 若$b>\varphi(p)$ 即使 a,p 不互质，$a^b=a^{b\bmod\varphi(p)+\varphi(p)}\pmod p$

2.欧拉降幂

利用以上公式可以求出 $k^{k^{k^{k\cdots}}}\pmod p$ 的结果

```
#include<bits/stdc++.h>
#define MAXN 10000100
#define LL long long
using namespace std;
const LL mod=1e9+7;
LL n,m,p,k;
bool is[MAXN];
LL prime[MAXN], cnt,phi[MAXN],sumphi[MAXN];
LL c[110],ccnt;
void init(int N){
    phi[1] = 1;
    is[1] = is[0] = 1;
    for(int i = 2; i <= N; i ++){
        if(!is[i]) prime[++cnt] = i, phi[i] = prime[cnt]-1;
        for(int j = 1; j <= cnt; j ++){
            if(i*prime[j] >= N) break;
            is[i*prime[j]] = 1;
            if(i % prime[j] == 0){
```

```
                phi[i*prime[j]] = phi[i] * prime[j];
                break;
            }else phi[i*prime[j]] = phi[i] * (prime[j] - 1);
        }
    }
    for(int i=1;i<=N;i++)sumphi[i]=(sumphi[i-1]+phi[i])%mod;
}

LL qpow(LL a,LL b,LL p)
{
    LL ans=1;
    for(a%=p;b;a=a*a%p,b>>=1)if(b&1)ans=ans*a%p;
    return ans;
}
LL Euler(LL k,LL p){return (p==1)?0:qpow(k,Euler(k,phi[p])+phi[p],p);}
//调用计算k^k^k^k...
int main()
{
    init(10000010);
    while(scanf("%lld%lld%lld",&n,&m,&p)!=EOF)
    {
        getc(n);
        printf("%lld\n",Euler(getk(ccnt,n,m),p));
    }
    return 0;
}
```

3.欧拉反演

$$n = \sum_{d|n} \varphi(d)$$

可以通过替换 n 来使用欧拉反演
比如用 $\gcd(i,j)$ 替换 n

$$gcd(i,j) = \sum_{d|i} \sum_{d|j} \varphi(d)$$

比如求

$$\sum_{i=1}^{n} gcd(i,n) = \sum_{i=1}^{n} \sum_{d|i} \sum_{d|n} \varphi(d) = n \sum_{d|n} \frac{\varphi(d)}{d}$$

可以认为欧拉反演是莫比乌斯反演的特化，以上内容都可以用莫比乌斯反演推导出来

蔡勒公式（日期转星期）

0-星期日，1-星期一，2-星期二，3-星期三，4-星期四，5-星期五，6-星期六

```
int Change(int year, int month, int day){
    if(month == 1 || month == 2) month += 12, year --;
    int c = year/100, y = year%100, m = month, d = day;
    int W = c/4 - 2*c + y + y/4 + 26*(m+1)/10 + d - 1;
    if(W < 0) return (W + (-W/7 + 1)*7)%7;
    return W % 7;
}
```