排列组合入门

1. 排列组合的基础知识

排列与组合

问题1：从1到9这九个数字中选三个数字出来组成一个三位数(其中不能有重复的数字)，问有多少种方案？

解： 方案数=9*8*7=504　　**排列问题！**

问题2：从学号为1到9这九同学中选三个同学出来打扫教室清洁，问有多少种方案？

解： 方案数=(9*8*7)/(3*2*1)=84　**组合问题！**

排列与组合

排列的定义：

从n个不同的元素中，取m个不重复的元素，按次序排列，称为从n个中取m个的排列。

方案数用A_n^m 来表示！（代码中写作A[n][m]）

$$A_n^m = \frac{n!}{(n-m)!}$$

例如：从n个不同的球中,取出m个,放入m个**不同**的盒子里，每盒1个。
方案总数为A[n][m]。

排列与组合

组合的定义：

从n个不同的元素中，取m个不重复的元素，不考虑次序，称为从n个中取m个的组合。

方案数用C_n^m 来表示！（代码中写作C[n][m]）

$$C_n^m = \frac{n!}{(n-m)!*(m!)}$$

例如：从n个不同的球中,取出m个,放入m个**相同**的盒子里，每盒1个。
方案总数为C[n][m]。

排列与组合

问题2 买糖果：

学校小卖部出售5种不同口味的糖果。何老板要买8颗糖，它希望每种口味的糖果都至少有一颗，问总共有多少种不同的购买方案？

此问题可反过来考虑，它等价于把8颗相同的糖果分配到5个不同的盒子里，求总的方案数。

我们考虑把这8颗糖果摆成一排，糖果间有7个空档。

我们在这7个空档中安插4块隔板，就能把糖果分成5份。

总的方案数为$C_7^4 = 35$

排列与组合

问题3 体操队形：

体操队7个同学站成一排，组成表演队形。其中4个男生，3个女生。教练要求，3个女生必须站在一起。问，有多少种不同的队形？

3个女生必须站在一起，我们可以考虑把她们看成1个人，加上4名男生，总共当作有5个同学。

5名同学的排列方案为A_5^5

其中代表女生的那个同学本身内部也有多种排列方案，方案数为A_3^3 。

根据乘法原理，总的方案数为$A_5^5 * A_3^3 = 720$

排列与组合

问题4 体操队形：

体操队10个同学站成一排，组成表演队形。其中7个男生，3个女生。教练要求，女生必须站在男生之间，且女生互不相邻。问，有多少种不同的队形？

从限制条件来看，男生限制少，女生限制多。我们考虑先排男生。

男生：7名男生的排列方案为A_7^7

女生：女生不能相邻。我们考虑把女生安插在男生之间。
7名男生之间有6个空档，选择其中3个空档来安插女生。共A_6^3 种方案。

根据乘法原理，总的方案数为$A_7^7 * A_6^3 = 604800$

组合数的性质

$$C_n^m = C_n^{n-m}$$

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

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

组合数的计算

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

```
//打表法,计算C[i][j] mod k
long long C[1000][1000];
for(int i=1;i<=n;i++)C[i][0]=1;

for(int i=1;i<=n;i++)
    for(int j=1;j<=i;j++)
        C[i][j] = (C[i-1][j]+C[i-1][j-1]) % k;
```

大组合数取模请参看后面的“Lucas定理”。

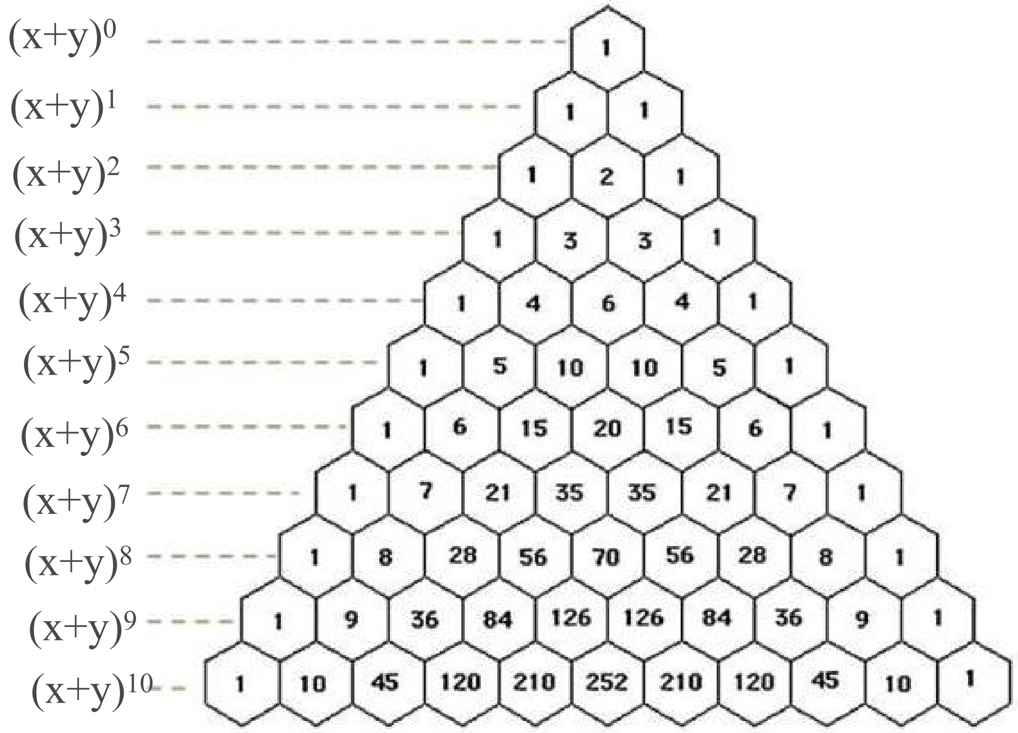
2. 二项式定理

二项式定理(牛顿二项式定理)

n是一个正整数。于是，对所有的x和y, 求 $(x+y)^n$ 展开后，每一项的系数

$$(x+y)^n = C_n^0 * x^n * y^0 + C_n^1 * x^{n-1} * y^1 + C_n^2 * x^{n-2} * y^2 + \ldots + C_n^{n-1} * x^1 * y^{n-1} + C_n^n * x^0 * y^n$$

即: $$(x+y)^n = \sum_{k=0..n} C_n^k * x^{n-k} * y^k$$



习题：NKOJ1327

3. Lucas定理

大组合数取模 Lucas定理

计算 C_n^m mod p　　　0<=m<=n<=100000，p是质数

Lucas定理：

定理1.

n、m是非负整数，p是**质数**,求C(n,m) mod p

把n和m写成p进制：

$m=a_k*p^k + a_{k-1}*p^{k-1} +...+ a_1*p^1 + a_0*p^0$

$n=b_k*p^k + b_{k-1}*p_{k-1} +...+ b_1*p^1 + b_0*p^0$

$C(n,m)\%p = C(b_k,a_k)*C(b_{k-1},a_{k-1})*...*C(b_1,a_1)*C(b_0,a_0)\%p$

例1.求C(22,10) MOD 3

$22=2*3^2+1*3^1+1*3^0$

$10=1*3^2+0*3^1+1*3^0$

C(22,10)%3 = C(2,1)*C(1,0)*C(1,1)%3

= 2*1*1 % 3 = 2

大组合数取模 Lucas定理

计算 C_n^m mod p　　　　0<=m<=n<=100000，p是质数

Lucas定理：

n、m是非负整数，p是**质数**，求C(n,m) mod p

Lucas(n,m,p)=C(n,m)%p

Lucas(n,m,p)=c(n%p,m%p)*Lucas(n/p,m/p,p)

例2.求C(22,10) MOD 3

```
Lucas(22,10,3) = C(22%3,10%3)*Lucas(22/3,10/3,3)
               = C(1,1)*Lucas(7,3,3)
               = C(1,1)*C(1,0)*Lucas(2,1,3)
               = C(1,1)*C(1,0)*C(2,1)%3
               = 2
```

Lucas定理 代码模板1

```
typedef long long LL;

LL C(LL a, LL b)
{
    if(a < b)   return 0;
    if(a == b)  return 1;
    if(b > a - b)   b = a - b;  //C(a,b)=C(a,a-b)

    LL A = 1, B = 1;
    for(LL i = 0; i < b; ++i)
    {                                   //C(a,b)=a!/((a-b)!*b!)
        A = (A * (a - i)) % p;   //计算a*(a-1)*...*(a-b+1) 即a!/(a-b)!
        B = (B * (b - i)) % p;   //计算b*(b-1)*...*1      即b!
    }//C(a,b)%p = (A/B)%p ,需要求B的逆元, p是质数, 根据费马小定理, 用快速幂求逆
    return (A * KSM(B,p-2,p)) % p;
}

LL Lucas(LL n, LL m)
{
    if(m == 0) return 1;
    return C(n % p, m % p) * Lucas(n / p, m / p) % p;
}
```

Lucas定理 代码模板2

```
LL KSM(LL a, LL b, LL p)   //二分快速幂求a^b mod p
{
    LL ans = 1;
    while(b)
    {
        if(b&1) ans = (ans * a) % p;
        a = (a*a) % p;
        b >>= 1;
    }
    return ans;
}
```

大组合数取模 Lucas定理

计算 $C_n^m \bmod p$

$0<=m<=n<=100000$

p是合数

需要用到"扩展Lucas定理"

详情自行百度！

4. 斯特林数
Stirling Numbers

第二类斯特林数

问题：何老板请客1 NKOJ4440

何老板在NK食堂订了m桌酒席，宴请信竞队的n名队员。由你来安排队员们就座。要求每桌至少坐1人。问总共有多少种安排方案？

第二类斯特林数$S_2[n][m]$表示把n个元素划分成m个非空集合的方案数。

$$S_2[n][m] = S_2[n-1][m-1] + m * S_2[n-1][m]$$

将第n个同学单独划入一个新的桌子中，即第n个同学独占一个桌子。前n-1个同学划分到m-1个桌子中。

前n-1个同学被分成了m个桌子。第n个同学加入到已经存在的m个桌子中，共m种选择。

第二类斯特林数 代码模板

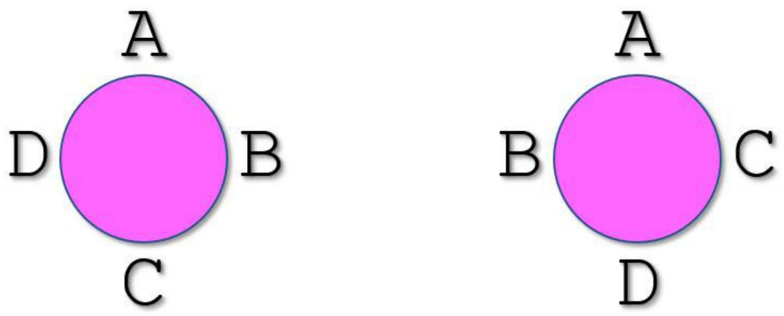
```
long long s2[maxn][maxn];      //存放要求的Stirling数
const long long mod=1e9+7;     //取模

void getStirling()
{
    for(i=1;i<=n;i++)s2[i][1]=1;
    for(int i=1;i<=n;i++)
        for(int j=2;j<=i&&j<=m;j++)
           s2[i][j]=(s2[i-1][j-1]+j*s2[i-1][j])%mod;
}
```

第一类斯特林数

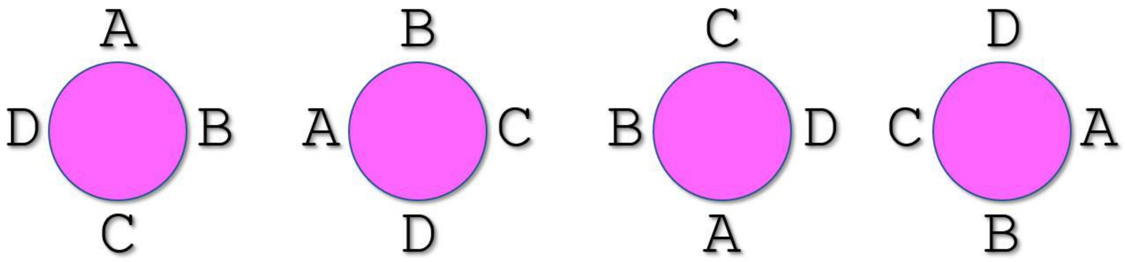
问题：何老板请客2 NKOJ4441

何老板在NK食堂订了m桌酒席，宴请信竞队的n名队员。酒桌为圆形。由你来安排队员们就座。要求每桌至少坐1人，最多坐n人。问总共有多少种不同的安排方案？

注意，同一桌队员就座的位置会影响方案数，比如A,B,C,D同桌，下面两种就座方案是不同的：



下面4种方案，我们认为它们是同一种方案：



第一类斯特林数

第一类斯特林数$S_1[n][m]$表示把n个元素划分成m个非空**循环排列**集合的方案数。

$$S_1[n][m] = S_1[n-1][m-1] + (n-1) * S_1[n-1][m]$$

将第n个同学单独划入一个新的桌子中，即第n个同学独占一个桌子。前n-1个同学划分到m-1个桌子中。

前n-1个同学被分成了m个桌子。把第n个同学安置到已经存在的n-1个同学种任意一个的左边即可。共n-1种方案。

第一类斯特林数 代码模板

```
long long s1[maxn][maxn];     //存放要求的Stirling数
const long long mod=1e9+7;    //取模

void getStirling()
{
    for(i=0;i<=n;i++)s1[i][i]=1;

       for(i=1;i<=n;i++)
         for(j=1;j<=m&&j<=i;j++)
           s1[i][j]=(s1[i-1][j-1]+(i-1)*s1[i-1][j])%mod;
}
```

BELL数

问题：何老板请客4

何老板在NK宴请信竞队的n名队员。要求每桌至少坐1人。问总共有多少种安排方案(注意没有规定有多少张桌子)？

BELL数B[n]表示把n个元素划分成若干个非空集合的方案数。

$$B[n] = S_2[n][1]+S_2[n][2]+S_2[n][3]+ \dots +S_2[n][N]$$

习题

NKOJ 1327 4051 4052

NKOJ 4440 4441 4442

NKOJ 4801