组合数学

1.排列组合

1.1 概述公式

排列：$P_n^m = \frac{n!}{(n-m)!}$

组合：$C_n^m = \frac{n!}{m!(n-m)!}$，后可能用$\begin{pmatrix} n \\ m \end{pmatrix}$代替

圆排列：$Q_n^m = \frac{P_n^m}{m} = \frac{n!}{m(n-m)!}$

多重集：包含重复元素的广义集合,$n = n_1a_1 + n_2a_2 + ... + n_ka_k$

多重排列数：也就是多重集的全排列，为$\frac{n!}{n_1!n_2!...n_k!}$

多重组合数1：$\forall i, m <= n_i$,即求$m = x_1 + x_2 + ... + x_k$的非负整数解，也就是$m + k = x_1 + x_2 + ... + x_m$的正整数解，然后必须插在中间，为$C_{m+k-1}^{k-1}$

多重组合数2：可以$m > n_i$，$ans = \sum_{p=0}^{k} (-1)^p \sum_A \begin{pmatrix} k + r - 1 - \sum\limits_A nA_i - p \\ k - 1 \end{pmatrix}$

不相邻排列：$1 \sim n$选k个数，任意两个不相邻的组合是$\begin{pmatrix} n - k + 1 \\ k \end{pmatrix}$

负数上指标组合数：$\binom{n}{m}(m < 0) = 0$；反转上指标，$\binom{n}{m}(n < 0) = (-1)^m \binom{m-n-1}{m}$

错位排列：$f(n) = (n-1)(f(n-1) + f(n-2))$

二项式定理：$(x + y)^n = \sum_{i=0}^{n} C_n^i x^i y^{n-i}$

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

$\sum_{i=0}^{m} C_n^i C_m^{m-i} = C_{m+n}^m$，$m = n$时,$\sum_{i=0}^{n} (C_n^i)^2 = C_{2n}^n$

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

$C_n^r C_r^k = C_n^k C_{n-k}^{r-k}$

$\sum_{i=0}^{n} C_{n-i}^i = F_{n+1}$，$F$是斐波那契数列

1.2 插板法：

应用条件：1.这n个元素必须互不相异；2.所分成的每一组至少分得1个元素；3.分成的组别彼此相异。

普通插板法：$x+y+z=10,x,y,z>0;$ 解数为 C_9^2

添元素隔板法：$x+y+z=10,x,y,z\geq 0;$ 转化为 $x+y+z=13,x,y,z>0;$ 解数为 C_{12}^2

添板插板法：
$x+y\leq 9,x,y>0;$ 添加一个板子 z 储存多余的元素，$z>=0;$ 转化为 $x+y+z>=10,x,y,z>0;$ 解数为 C_9^2

选板法：有10颗糖，每天至少吃一颗，有几种吃法，1_1_1_1_1_1_1_1_1_1，1为糖，_为板，有2^9种

```
ll C[2010][2010];//1
void get_C(int maxn){
    C[0][0]=1;
    for(int i=1;i<=maxn;i++){
        C[i][0]=1;
        for(int j=1;j<=i;j++)
            C[i][j]=(C[i-1][j]+C[i-1][j-1])%mod;
    }
}
int fac[N],inv[N];//2
int qpow(int a,int n){
    int ret=1;
    while(n){
        if(n&1) ret=ret*a%mod;
        a=a*a%mod;
        n>>=1;
    }
    return ret;
}
void init(int n){
    fac[0]=1;
    for(int i=1;i<=n;i++)fac[i]=fac[i-1]*i%mod;
    inv[n]=qpow(fac[n],mod-2);
    for (int i=n-1;i>=1;i--)inv[i]=inv[i+1]*(i+1)%mod;
}
int C(int x,int y){
    if (x<y)return 0;
    return fac[x]*inv[y]%mod*inv[x-y]%mod;
}
ll F[100010];//3 卢卡斯定理
void init(ll p){
    F[0] = 1;
    for(int i = 1;i <= p;i++)
        F[i] = F[i-1]*i % (1000000007);
}

ll inv(ll a,ll m){
    if(a == 1)return 1;
    return inv(m%a,m)*(m-m/a)%m;
}

ll Lucas(ll n,ll m,ll p){
    ll ans = 1;
    while(n&&m){
        ll a=n%p;
```

```
        ll b=m%p;
        if(a<b)return 0;
        ans=ans*F[a]%p*inv(F[b]*F[a-b]%p,p)%p;
        n/=p;m/=p;
    }
    return ans;
}
```

1.3 二项式定理和二项式反演

二项式定理：

$$(x+y)^n=\sum_{i=0}^{n}C_n^i x^i y^{n-i},\ n\ge 0$$

$$\text{例题：}1.\sum_{i=0}^{n}\binom{n}{i}i=\sum_{i=0}^{n}\frac{n}{i}\binom{n-1}{i-1}i=n\sum_{i=0}^{n}\binom{n-1}{i-1}=n\sum_{i=0}^{n-1}\binom{n-1}{i}=n2^{n-1}$$

$$2.\sum_{i=0}^{n}\sum_{j=0}^{i}\binom{n}{j}=\sum_{j=0}^{n}\binom{n}{j}(n-j+1)=(n+1)\sum_{j=0}^{n}\binom{n}{j}-\sum_{j=0}^{n}\binom{n}{j}j=(n+1)2^n-n2^{n-1}=(n+2)2^{n-1}$$

$$3.\sum_{i=0}^{n}i^2C_n^i=\sum_{i=0}^{n}i(i-1)C_n^i+\sum_{i=0}^{n}iC_n^i=n(n-1)\sum_{i=0}^{n-2}\binom{n-2}{i}+n\sum_{i=0}^{n-1}\binom{n-1}{i}=n(n+1)2^{n-2}$$

$$4.\sum_{i=k}^{n}\binom{i}{k}=>\sum_{i=k}^{n}(x+1)^i\text{的}k\text{次幂系数}$$

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

$$k\text{次幂系数为}\binom{n+1}{k+1}$$

广义二项式定理

$$(x+y)^n=\sum_{i=0}^{n}C_n^i x^i y^{n-i}$$

$$n\text{可以为任意实数和复数，只需}|x/y|<1\text{保障收敛}$$

二项式反演(类似容斥)

$$\text{对于两个数列}f,g\text{有}$$

$$f_n=\sum_{i=0}^{n}(-1)^i\binom{n}{i}g_i\iff g_n=\sum_{i=0}^{n}(-1)^i\binom{n}{i}f_i$$

$$\text{变形有}f_n=\sum_{i=0}^{n}\binom{n}{i}g_i\iff g_n=\sum_{i=0}^{n}\binom{n}{i}f_i$$

$$\text{例子：}1.\text{恰好和至多：错排问题},\text{设}f_i\text{表示恰好有}i\text{个错开，设}g_i\text{表示至多有}i\text{个错开，}g_i=i!\text{，求}f_n$$

$$2.\text{恰好和至少：}$$

$$BZOJ2839\text{集合计数：一个有}N\text{个元素的集合有}2^N\text{个不同子集，现在要在这}2^N\text{个集合中取出若干集合（至少一个），}$$

$$\text{使得它们的交集的元素个数为}K\text{，求取法的方案数}$$

$$\text{设}f_i\text{表示交集恰好有}i\text{个元素，设}g_i\text{表示至少有}i\text{个元素，}g_i=\binom{n}{i}(2^{2^{n-i}}-1)$$

$$(\text{先确定}i\text{个要选的，再将其他元素先拼成集合，再把集合拼成集合，且一定要选一个集合})$$

1.4 容斥原理

$$|\cup_{i=1}^{n} A_i| = \sum_{T \subseteq \{1,2,\ldots,n\}} (-1)^{|T|-1} |\cap_{j \in T} S_j|$$

$$例：|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$$

$$证明：p在k个元素中出现过，则p对答案的贡献为，i为其中i个元素的集合，\sum_{i=1}^{k} (-1)^{i-1} \binom{k}{i} = 1 - \sum_{i=0}^{k} (-1)^{i} \binom{k}{i} = 1 - (1-1)^k = 1$$

$$推出P(A_i)为A_i发生的概率，P(\cup_{i=1}^{n} A_i) = \sum_{T \subseteq \{1,2,\ldots,n\}} (-1)^{|T|-1} P(\cap_{j \in T} S_j)$$

$$例题：硬币购物4种硬币c1, c2, c3, c4，n次询问，每次给定四种硬币个数D_i和付款金额S，共有多少支付方式，n \leq 10^3, S \leq 10^5$$

$$n次多重背包，用单调队列优化，最优复杂度O(n4S)$$

$$完全背包+容斥，满足D_i限制的方案数=全部方案数-不满足的方案数，完全背包O(4S)预处理dp[S]$$

$$不满足的情况|\cup_{i=1}^{4} A_i|，|A_i|为dp[S - C_i(D_i + 1)]$$

1.5 错排问题

介绍：n个邮箱编号1-n，n份信编号1-n，每份信都装到相同编号的邮箱内，每个邮箱只收一份信，求所有信都装错的情况数

$$f(n) = (n-1)(f(n-1) + f(n-2))$$

$$通项公式：设A_i为第i份信送到第i个邮箱内的情况数，A_i = (n-1)!$$

$$A_i \cap A_j = (n-2)!, k个A集合交 = (n-k)!$$

$$f(n) = n! - |\cup_{i=1}^{n} A_i| = n! - \sum_{i=1}^{n} (-1)^{i-1} \binom{n}{i} (n-i)! = n! - \sum_{i=1}^{n} (-1)^{i-1} \frac{n!}{i!}$$

2.卡特兰数

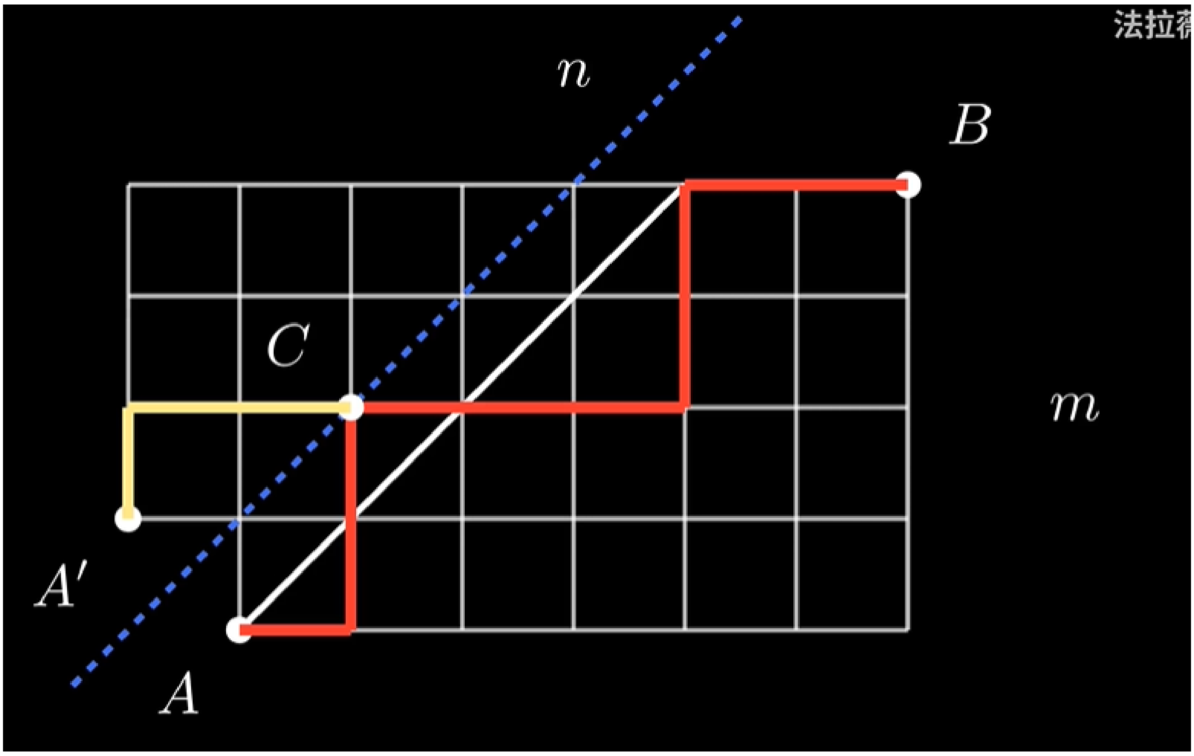
$H_n = \frac{C_{2n}^{n}}{n+1}$

$H_n = \frac{H_{n-1}(4n-2)}{n+1}$

$H_n = C_{2n}^{n} - C_{2n}^{n+1}$

扩展卡特兰数：$C_{n+m}^{n} - C_{n+m}^{n+1}$

折线法：A到B的全部路线-经过蓝线的所有路线



3.斯特林数

斯特林公式：$n! \approx \sqrt{2\pi n}(\frac{n}{e})^n$

第二类斯特林数：将n个两两不同的元素，划分为m个互不区分的非空子集的方案数。

$S(n,m)=m*S(n-1,m)+S(n-1,m-1),S(n,0)=[n==0]$

通项公式：

$$S(n,m)=\sum_{i=0}^{m}\frac{(-1)^{m-i}i^n}{i!(m-i)!}$$

4.母函数与生成函数

组合：$a_0+a_1x+a_2x^2+a_3x^3+\ldots$

排列：$a_0+\frac{a_1x}{1!}+\frac{a_2x^2}{2!}+\frac{a_3x^3}{3!}+\ldots$

卷积：

范德蒙德卷积：$(1+x)^{s+n}$的x^m系数,$\sum_{i=0}^{m}\binom{n}{i}\binom{s}{m-i}=\binom{s+n}{m}$

$(1-x^2)^r$的x^n系数,n为奇数,系数为0;n为偶数,系数为 $\sum_{i=0}^{n}\binom{r}{i}\binom{r}{n-i}(-1)^i=(-1)^{n/2}\binom{r}{n/2}$

特殊恒等式

$$\frac{1}{(1-z)^{n+1}}=(1-z)^{-n-1}=\sum_{k}\binom{-n-1}{k}(-z)^k=\sum_{k\geq 0}(-1)^k\binom{k-(-n-1)-1}{k}(-z)^k=\sum_{k\geq 0}\binom{k+n}{k}z^k$$

$$\frac{z^n}{(1-z)^{n+1}}=\sum_{k\geq 0}\binom{k+n}{k}z^{k+n}=\sum_{k\geq 0}\binom{n}{k}z^k(\text{令}\,k=k+n$$

$$n=0\text{时，}\frac{1}{1-z}=\sum_{k\geq 0}z^k$$

$$\text{序列}<1,1,\ldots,1>\text{和}<a_n>\text{的卷积},c_n=\sum_{i=0}^{n}a_i$$

5.康托展开

```
//O(n^2)排列是第几个
#include<bits/stdc++.h>
using namespace std;
int n,m,f[11],a[11];
char ka[11];
vector<int>v;
inline int cantor(){
    int ret=0,x;
    for(int i=0; i<n; ++i) {
        x=0;
        for(int j=i+1; j<n; ++j)
        if( (ka[i]-ka[j])>0 ) x++;
        ret+=x*f[n-i-1];
    }
    return ret;
}
inline void incantor(int k){
    int x;
    while( !v.empty() ) v.erase(v.end());
    for(int i=1; i<=n; ++i) v.push_back(i);
    for(int i=1; i<n; ++i) {
        a[i]=v[(x=k/f[n-i])];
        v.erase(v.begin()+x);
        k%=f[n-i];
    }
    a[n]=v[0];
}

int main(){
    cin>>n;
    for(int i=0; i<n; ++i) cin>>ka[i];
    f[1]=1;
    for(int i=2; i<=10; ++i) f[i]=f[i-1]*i;
    incantor(cantor()-1);
    for(int i=1; i<=n; ++i) printf("%d ",a[i]);
    return 0;
}
```