# 群论

## 置换群

Poj 1026  
题意：给出一个n个数的置换，按照置换规则把一个字符串置换k次，如果字符串的长度不足n，则在字符串末尾补空格，直到长度为n。求置换k次之后的字符串是什么。

分析：如果直接按照题目描述的模拟，肯定会超时。  
对整个字符串置换，可以转换为对每个循环节进行置换。因为一个循环节里面的元素,对其他元素没有影响，这样我们就可以先求出所有的循环节和每个循环节中的元素及循环节的长度，然后对每个循环节中的元素进行置换。一个循环节中的元素置换k次，等价于每个元素置换k%（这个元素所在循环节的长度）次，这样模拟次数变得很小，就可以模拟了。

Poj 3270

现有一排装满货物的箱子，重量各不相同（都已标示在箱子上），为了进行后面的工作，需要将这些箱子按轻重有序放置，但只有一名工作人员来完成这项工作，由于空间有限，他只能通过不断交换两个箱子（可不相邻）的位置的方式来实现箱子的排序。他知道这样一定可以完成任务，但搬箱子很累，所以他希望找到一种最省力的方式来完成工作，假设一次交换两个箱子的代价为这两个箱子的重量之和，那么这项工作的总代价为此过程中所有“交换”的代价之和，请帮助他计算排列这些箱子的最小代价。

做完这道题目看网上题解时候发现有一篇题解指出：涉及可以不相邻的交换代价的题目一般都与置换群有关。也许这是久经沙场的ACM/OI老将的经验直觉，这个题目虽然可以看作置换群，但是一开始想到置换群的圈结构是不容易的，而且过分依赖于经验直觉直接认为它是置换群的题目再去凑解法也是不自然的，所以我们要从头开始分析这个题目。

首先，考虑到两个因素可以是代价小：1.尽量移动小元素做中转。2.如果把总置换看作一次一次置换的乘法，尽量不要牵涉多余的元素进入一次置换。   
比如(300 400 500 200)->(200 300 400 500)这个一次置换，在这两个因素指导下，尽量在内部换，而且尽量多与200交换。那么，可以得到一个代价200\*3+300+400+500，似乎问题得到了解决，可以发现如果严格遵守2.，一个长为n的圈A的最代价为minA \* (n-2)+sum A，由置换群理论，所有的置换都可以分解为圈。   
但是这样一来如果允许外面元素交换入内，比如外面存在一个全置换群最小元素，比如1，与200换后，事实上可以减小代价。这也是应当考虑的，那么，如果可以交换进来，是不是只能交换一个全置换群最小元素呢？是正确的，因为200的角色只需要一个元素扮演，那么经过代数推算发现是否应该交换与200和1的相对大小有关，所以如果全置换群最小元素为m，一个圈A的最小代价为min cost A=sumA+min{minA\*(n-2)，minA+(n+1) \* m}。   
这就是置换群+贪心[**算法**](http://lib.csdn.net/base/datastructure)的由来，虽然细节没有证明，但是证明基本严谨。

Polya定理的应用。先来看Polya定理。

Polya定理：设 G = {a1，a2，…，ag}是 N 个对象的置换群，用 M 种颜色给这 N 个对象着色，则不同的着色 方案数为：

                  |G|^(-1) \* {M^c(a1) + M^c(a2) + … + M^c(ag)}。

置换：比如反转，旋转，对折的一种方案就是一个置换

其中 c(ai)为置换 ai 的循环节数，( i = 1，2，…，g )。

如（1）（3，4，5）是一个长度为1的循环节和一个长度为3的循环节

其中|G|=g是所有置换的个数

## Poj 2409

题目大意：

给定M种颜色的珠子，每种颜色珠子的个数均不限，将这些珠子做成长度为N的项链。

问能做成多少种不重复的项链，最后的结果不会超过int类型数据的表示范围。并且两条项链相同，当且仅当两条项链通过旋转或是翻转后能重合在一起，且对应珠子的颜色相同。

对于这道题，直接用Polya定理求解，找出所有的置换，并求出置换的循环节数。然后根据上边公式求出 M^c(ai) 的总和，再除以置换群个数。

题中有两种置换方式：

1.旋转置换。分别顺时针旋转 i 个珠子，其循环节长度为 LCM(N，i) / i，循环节数为

N / (LCM(N，i) / i)，即 GCD(N，i)。

2.翻转置换。根据 N 的奇偶性分情况讨论。

N为奇数时：

       以第 i 个珠子为顶点和中心翻转，翻转后，第 i 个珠子保持不变，其余珠子两两相

互对换，因为有 N 个珠子，所以有 N 种翻转置换，每种翻转循环节数为 (N+1) / 2。

N为偶数时，有两种翻转方式：

      以两边相对的两个珠子为轴和中心翻转，翻转后，这两个珠子保持不变，其余珠子

两两相互对换，共有 N/2 种翻转置换，每种翻转循环节数为 (N+2) / 2。

      以相邻的珠子中间连线为轴和中心翻转，翻转后，所有珠子两两相互对换，共有 N/2

种翻转置换，每种翻转循环节数为 N/2。

所以无论奇偶总有2N个置换，然后根据Polya定理求出结果。

poj 2154：

这题和2409 类似，不同之处在于，只考虑旋转，不考虑翻转；因此相对前面两个题目应该说是更简单，但一看数据范围，就不是这么回事了，2409完全可以直接循环处理，但这题目n最大达100000000，显然会TLE，故需寻求更佳的解决方案。用欧拉函数进行优化：

旋转：顺时针旋转i格的置换中，循环的个数为gcd（i，n），每个循环的长度为n/gcd(i,n)。

如果枚举旋转的格数i，复杂度显然较高。有没有好方法呢？可以不枚举i，反过来枚举L。

由于L|N,枚举了L，再计算有多少个i使得0<=i<=n-1并且L＝gcd(i, n)。即gcd（i,n）＝n/L。

不妨设a＝n/L=gcd(i, n),

不妨设i＝a\*t则当且仅当gcd(L,t)=1时

Gcd(i,n)=gcd(a\*L,a\*t)=a。

因为0<=i<n,所以0<=t<n/a=L.

所以满足这个条件的t的个数为Euler(L).

burnside引理

http://wenku.baidu.com/link?url=1geirKBzAaO8NeoXJzKYz6Ei3jf5XMPqu4ZEWUPJXQP7udrZsgEQp990URCve2xIy56UrsMp32ewYHuz63E-qSGFpoJiewgyiQkZeXHfXhC

burside不如polya快捷但可以解决有限制条件（如某两种颜色珠子不相邻）的限制性染色问题.

设G={a1,a2,…ag}是目标集[1,n]上的置换群。每个置换都写成不相交循环的乘积。

http://g.hiphotos.baidu.com/baike/s%3D43/sign=ec314b49718da9774a2f8728b151a39e/5243fbf2b211931316df77ca67380cd791238d28.jpg

是在置换ak的作用下不动点的个数，也就是长度为1的循环的个数。通过上述置换的变换操作后可以相等的元素属于同一个[等价类](http://baike.baidu.com/view/732593.htm)。若G将[1,n]划分成l个等价类，则：

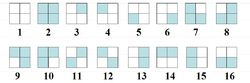
等价类个数为：

http://e.hiphotos.baidu.com/baike/s%3D249/sign=4819311e96eef01f49141fc1d9ff99e0/94cad1c8a786c91783a8da43cb3d70cf3bc75725.jpg

对于置换理解为所有涂色方案的图（如下图16种）经过变换（如下图有4种）后图与图之间的关系，若图1通过该种变换变为图3，而图3又可以经过此变换变为图2，图二经过此变换变为图1，即有关系1-》3-》2-》1则成为一个循环(132)，不动点即为长度为1的循环。

**例1：**一正方形分成4格，2着色，有多少种方案？其中，经过转动相同的图象算同一方案。

**解：**每个格子一共有两种颜色可以选择，所以共有右图16中图像。

[](http://baike.baidu.com/pic/burnside%E5%BC%95%E7%90%86/1505996/0/b58f8c5494eef01fee7f92bfe2fe9925bc317d65?fr=lemma&ct=single)

对图中图像的置换可以分为以下四种：

不动：a1=(1)(2)…(16)

逆时针转90度 ：a2=(1)(2)(3 4 5 6)(7 8 9 10) (11 12)(13 14 15 16)

顺时针转90度 ：a3=(1)(2)(6 5 4 3)(10 9 8 7)(11 12)(16 15 14 13)

转180度：a4=(1)(2)(3 5)(4 6)(7 9)(8 10)(11)(12) (13 15)(14 16)

由Burnside引理，共有(16+2+2+4)/4=6(种方案)

由例子可见，Burnside引理是针对图像集的转动群来求解，当多种颜色着色时，理论上可以用Burnside来求解，但是极其复杂，此时一般通过Pólya定理求解。

Poj 2888(折磨一下午)

http://blog.csdn.net/raalghul/article/details/51767941

一个长度为N(N<=1000000000)的环形珍珠项链，有M(M<=10)种颜色，每个珍珠可以任选一种颜色染色，但是规定了K对禁止关系(a,b)，即不能存在颜色分别为a,b的珍珠相邻，询问可以有多少种染色方法可以得到本质不同的珍珠项链(通过旋转得到的项链为本质相同)

题解1：我们来回顾一下Pólya原理的推出，有一个神奇的Burnside引理指出：一个本质不同的组合数等于所有置换下不动点的个数之和除以置换种类数，针对此题我们不妨从Burnside引理挖掘一下思路，虽然Pólya原理计算很快，但是却无法应用，如果退一步直接用Burnside引理半暴力计算可能就是可以的了。于是我们我们不妨设L为置换错位的个数，所有的置换就是L={0,1,2,3……,N-1}，我们可以分别求一下每个L的不动点个数，有一个结论：在L置换下，置换中循环节相邻两个元素距离相同，且都为gcd(L,N)（证明同鞋可以自行脑补），所以求L置换的不动点只需求长度为gcd(L,N)的所有染色方案数（此处要保证gcd(L,N)号珍珠能和1号珍珠连接，因为N是由一堆长度为gcd(L,N)的链连接起来的）如果用0,1矩阵A来表示无向图的连通情况的话，A^k代表的就是一个点经过k条路后能到达的地方的方法数。

因此，对于循环节为r的情况，A^r就是任意点经过r条路能到达的地方，与之对应的map[i][i]就是一个珠子经过可行路径转了r条路径又回到自己的种数，其实，就是前面说的满足题意的排列数，

题解2：一看到本质不同，第一反应就是Pólya原理计数，但是因为有K对限制，所以每种置换不动点不是（m^循环节个数），直接用Pólya显然是不可做的。所以我们想办法从Pólya原理的得出入手。

。于是此题经过Burnside引理转化为了一道链上的组合计数问题，可以考虑用动态规划解决。

动态规划的解决方法：由于限制只有相邻两位，我们不妨设f[i][j]表示递推到第i位，最后一位颜色为j的方案数，我们可以先虚拟规定第0位的颜色为A[0]，然后向后递推到gcd(L,N)后取第gcd(L,N)位=A[0]时的方案数。

第一步优化：如果对于1~N-1的所有L都动规一遍显然是大大超时，我们观察得到，对于很多L，gcd(L,N)是相同的，于是我们只需做一遍gcd(L,N)的动规就可以得到所有gcd(i,N)=gcd(L,n)（1<=i<=N）的方案数，此处用到了一个结论，i∈1~N所有gcd(i,N)=k i的个数=φ(N/k)（φ(x)=x的欧拉函数），于是我们就可以循环i到sqrt(N)对每个i|N的i做一次动规就行了。

第二步优化：由于N非常大而M非常小于是O(NM)的动规显然是不行的，观察到动规方程是单纯的求和计数转移，于是我们很自然的联想到矩阵优化动规O(log(N)M^3)，Matrix[i][j]=1当且仅当颜色i与j能相连。做长度为K的动规时，方案数就等于Matrix^K对角线上所有元素的和。不妨设第i行行向量表示A[0]=i的状态，初始就是单位矩阵，根据矩阵乘法原理每右乘一次Matrix相当于每行向量在Matrix上的转移，所以Matrix^i的第k行第j个数就相当于A[0]=k时的f[i][j]。

最后因为Burnside引理最后要除N，所以最后答案乘一下N在mod=9973下的逆元就行了。