ACM中的数学问题

北京大学ACM竞赛队队员
林舒

引言

- 在ACM竞赛中,经常可以看到数学问题的身影
- 可以是纯数学问题,也可以是需要利用数学上的一些公式,定理,算法来辅助解决的问题
- 会者不难,而不会的人在赛场上一般很难推出公式或进行证明
- 往往想起来费劲,写起来却很轻松

常见的数学问题

- 数论
- 组合数学
- 博弈论
- 线性代数
- 高等数学
- 线性规划
- 概率论
- …

本讲内容

- 简单数论
- Polya定理
- SG函数
- 与矩阵有关的问题

本讲内容

- 基本上是最基础的,同时也是ACM竞赛中最常用的数学算法
- 一些数学定理,公式的简略证明或推导,从而加深对它们的理解和认识,也方便记忆
- 往届ACM竞赛中的数学问题

数论

- 简而言之,数论是研究整数的理论
- 在ACM竞赛中,经常用到数论的相关知识
- 纯数论的题目不多,大部分是和其他类型的问题结合起来的
- 约数,倍数,模线性方程,欧拉定理,素数

整除的性质

- 性质1: $a|b, b|c \Rightarrow a|c$
- 性质2: $a|b \Rightarrow a|bc$
- 性质3: $a|b, b|c \Rightarrow a|kb \pm lc$
- 性质4: $a|b, b|a \Leftrightarrow a = \pm b$
- 性质5: $a = kb \pm c \Leftrightarrow a,b$ 的公因数与 b,c 的公因数完全相同(利用性质3)

最大公约数
最小公倍数

- 欧几里德算法(辗转相除法,短除法)
- 原理:若$a \equiv r \pmod{b}$,则$\gcd(a,b)=\gcd(b,r)$(利用性质5证明)
- 算法步骤(递归实现):
  - 整数a,b,假设a>b
  - 若b=0,则gcd(a,b)=|a|;否则gcd(a,b)=gcd(b,a%b)
- 最小公倍数:lcm(a,b)=a*b/gcd(a,b)

解二元模线性方程

- 二元模线性方程(二元一次不定方程):形如 $ax\equiv c \pmod b$ 或 $ax+by=c$
- 扩展欧几里德算法
- 原理:
  - 令 $d=\gcd(a,b)$,原方程有整数解当且仅当 $d|c$
  - $bx+(a\%b)y=1 \Leftrightarrow ay+b(x-[a/b]*y)=1$

解二元模线性方程

- 算法步骤:
  - 整数a,b,c,设d=gcd(a,b)
  - 在用欧几里德算法求gcd(a,b)的过程中求方程ax+by=d的一组整数解:
    - 若b=0,则x=1,y=0;
    - 否则,递归调用gcd(b,a%b),可以得到bx'+(a%b)y'=d的解x',y',令x=y',y=x-[a/b]y'即满足ax+by=d
  - 若d|c,设c=kd,则有a(kx)+b(ky)=c;否则原方程无整数解

中国剩余定理

- 孙子定理,韩信点兵,隔墙算,鬼谷算,大衍求一术...
- "物不知数"问题:"今有物不知其数,三三数之剩二,五五数之剩三,七七数之剩二,问物几何?答曰:'二十三.'术曰:三三数之剩二,置一百四十,五五数之剩三,置六十三,七七数之剩二,置三十,并之,得二百三十三,以二百一十减之,即得.凡三三数之剩一,则置七十,五五数之剩一,则置二十一,七七数之剩一,则置十五,即得." --孙子算经

中国剩余定理

- 一般性问题:给定两两互质的正整数$n_1,n_2,\ldots,n_k$,要求找到最小的正整数a,满足$a\equiv a_i \pmod{n_i}$
- 算法步骤:
  - 令$n=n_1n_2\ldots n_k, m_i=n/n_i$
  - 显然$\gcd(m_i,n_i)=1$,利用扩展欧几里德算法计算出x_i满足$m_ix_i\equiv 1 \pmod{n_i}$
  - $a\equiv a_1x_1m_1+a_2x_2m_2+\ldots+a_kx_km_k \pmod{n}$

筛法

- 目标:求出n以内的所有质数
- 算法步骤:
  - 初始时容器内为2到n的所有数
  - 取出最小的数p,p一定是质数,删去p的所有倍数(注:只需从p^2开始删除即可)
  - 重复上述步骤直到容器为空
  - 用bool数组实现即可
- 缺陷:一个数可能被重复删去多次,影响效率

筛法

- 改进:
  - 初始时容器内为2到n的所有数
  - 取出最小的数p,p一定是质数
  - 删去所有的p^i,令q为第一个未被删除的数,保留q,删去所有的p^iq,再令q为下一个未被删除的数,删去所有的p^iq...直到q遍历所有未被删除的数为止(这一步骤可以把最小质因数为p的所有数删去)
  - 重复上面两个步骤直到容器为空
  - 用bool数组+双向链表实现
- 小优化:初始时只加入奇数

算术基本定理

- 任何一个大于1的自然数n,都可以唯一分解成有限个质数的乘积
- $n=p_1^{r1}p_2^{r2}\ldots p_k^{rk}$
- $p_1<p_2<\ldots<p_k$均为质数,$r_1,r_2,\ldots r_k$均为正整数

欧拉函数

- 记φ(x)为与x互质且小于x的正整数个数
- 设$x=p_1^{r1}p_2^{r2}\ldots p_k^{rk}$
- $\varphi(x)=x*(1-1/p_1)*(1-1/p_2)*\ldots*(1-1/p_k)$或
  $\varphi(x)=p_1^{(r1-1)}(p_1-1)p_2^{(r2-1)}(p_2-1)\ldots p_k^{(rk-1)}(p_k-1)$
- 递推式:质数p满足$p|x$,若$p^2|x$,则
  $\varphi(x)=\varphi(x/p)*a$,否则$\varphi(x)=\varphi(x/p)*(a-1)$

欧拉函数

- 证明：
  - 若p为质数，则$\varphi(p)=p-1$
  - $\varphi(p^r)=p^r(1-1/p)=p^{(r-1)}(p-1)$
  - 若a,b互质，则$\varphi(ab)= \varphi(a)\varphi(b)$
- 扩展：n的所有因子之和

$$(p_1^0+\ldots+p_1^{r1})(p_2^0+\ldots+p_2^{r2})\ldots(p_k^0+\ldots+p_k^{rk})$$

欧拉定理

- 若a和m互质,则$a^{\varphi(m)} \equiv 1 \pmod{m}$
- 证明:
  - 设$\varphi(m)$个正整数$r_1, r_2, \ldots, r_{\varphi(m)}$满足:$r_i$与m互质,对于任意$i \neq j$,$r_i \neq r_j \pmod{m}$
  - 利用反证法可以简单证明$ar_1, ar_2, \ldots, ar_{\varphi(m)}$依然满足上述条件
  - $(ar_1)(ar_2)\ldots(ar_{\varphi(m)}) \equiv a^{\varphi(m)} r_1 r_2 \ldots r_{\varphi(m)} \pmod{m}$
  - 两边同时约去$r_1 r_2 \ldots r_{\varphi(m)}$即得到欧拉定理

素数测试

- 费马小定理:若p为素数,则对于任意小于p的正整数a,有$a^{(p-1)} \equiv 1 \pmod p$
- 证明:用欧拉定理直接得出
- 二次探测定理:若p为素数,$a^2 \equiv 1 \pmod p$小于p的正整数解只有1和p-1
- 满足费马小定理和二次探测定理的数可以确定是素数

素数测试

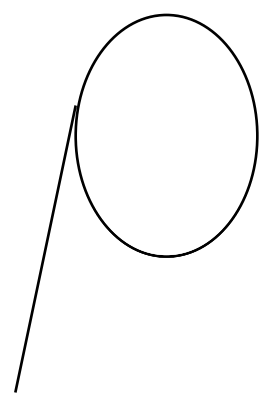
- Miller-Rabin算法
- 算法步骤：
  - 判定n是否为素数
  - 令$n-1=m*2^j$,m为奇数
  - 随机在2到(n-1)之间取一个整数b
  - 令$v=b^m$,之后每次对v平方,当v=1时,若上一次的v既不是1也不是(n-1),由二次探测定理,n不是素数,退出;不断循环直到计算出$b^{(n-1)}$
  - v=1,满足费马小定理,通过测试;否则n一定不是素数
  - 选取几个不同的b多次测试

素数测试

- Miller-Rabin只能算一种测试,因为通过测试的数不一定是素数,非素数通过测试的概率是1/4
- 虽然一次测试的结果不一定令人满意,但五六次随机测试基本可以保证正确率超过99.9%

大整数分解

- 至今仍是世界难题
- 在密码学中起着至关重要的作用
- 试除法,Fermat方法,Pollard方法
- Pollard rho方法



Pollard rho方法

- 原理:用某种方法生成两个整数a和b,计算p=gcd(a-b,n),直到p不为1或a,b出现循环为止,若p=n,则n为质数,否则p为n的一个约数
- 算法步骤:
  - 选取一个小的随机数x_1
  - 迭代生成$x_i=x_{(i-1)}^2+k$,一般取k=1,若序列出现循环则退出
  - 计算$p=gcd(x_{(i-1)}-x_i,n)$,若p=1,返回上一步;直到p>1为止
  - 若p=n,则n为素数,否则p为n的一个约数并递归分解p和n/p

数论小结

- 前面的这些都是一些初等数论的知识
- 可以看出,数论所研究的内容,很大一部分都是和素数紧密联系的.因此,数论是名副其实的"素论"
- 这些算法代码量都不大,但如果没有准备好模版的话,往往会忽略一些细节

Polya定理

- 组合数学理论中最重要的定理之一
- 在组合计数问题中有重要作用
- 涉及的概念和定理比较多,证明较复杂,本讲只是粗略地介绍

一个经典的例子

- 用两种颜色去染排成一个圈的6个棋子，如果能够通过旋转得到只算作一种,问有多少种染色状态
- 下面将通过这个例子来形象地介绍Polya定理的内容和解决这类问题的方法

置换

- 置换:用矩阵形式表示的顶点的变换
- 例子中,将棋子从某个点顺时针标上1到6,则将所有棋子顺时针旋转一个位置的置换可表示为:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 6 & 1 & 2 & 3 & 4 & 5 \end{pmatrix}$$

置换群

- 以置换为元素的群
- 置换群$G=\{a_1,a_2,\ldots,a_{|G|}\}$
- 例子中G内共有6个置换

$$\begin{pmatrix}1&2&3&4&5&6\\1&2&3&4&5&6\end{pmatrix}\quad\begin{pmatrix}1&2&3&4&5&6\\6&1&2&3&4&5\end{pmatrix}\quad\begin{pmatrix}1&2&3&4&5&6\\5&6&1&2&3&4\end{pmatrix}$$

$$\begin{pmatrix}1&2&3&4&5&6\\4&5&6&1&2&3\end{pmatrix}\quad\begin{pmatrix}1&2&3&4&5&6\\3&4&5&6&1&2\end{pmatrix}\quad\begin{pmatrix}1&2&3&4&5&6\\2&3&4&5&6&1\end{pmatrix}$$

循环

- 在一个置换下,x_1->x_2,x_2->x_3,…,x_n->x_1,这样x_1,x_2,…,x_n就构成了一个循环
- 定义c_k为在置换a_k下的循环总数
- 例子中:

  $c_1=6, c_2=1, c_3=2, c_4=3, c_5=2, c_6=1$

Polya定理

- 设$G=\{a_1,a_2,\ldots,a_{|G|}\}$是$N=\{1,2,\ldots,N\}$上的置换群,现用m种颜色对这N个点染色,则不同的染色方案数为
  $S=(m^{c_1}+m^{c_2}+\ldots+m^{c_{|G|}})/|G|$
- 证明比较复杂,略

利用Polya定理解决组合计数问题的步骤

- 写出置换群

$$\begin{pmatrix}123456\\123456\end{pmatrix}\quad\begin{pmatrix}123456\\612345\end{pmatrix}\quad\begin{pmatrix}123456\\561234\end{pmatrix}$$

$$\begin{pmatrix}123456\\456123\end{pmatrix}\quad\begin{pmatrix}123456\\345612\end{pmatrix}\quad\begin{pmatrix}123456\\234561\end{pmatrix}$$

- 求出每个置换的循环数

  $c_1=6, c_2=1, c_3=2, c_4=3, c_5=2, c_6=1$

- 计算染色方案

  $S=(2^6+2^1+2^2+2^3+2^2+2^1)/6=14$

常见置换的循环数

- 计算置换的循环数,是这一算法的瓶颈.如果能够快速计算出各置换的循环数,就可以大大提高程序的运行效率
- 旋转:n个点顺时针(或逆时针)旋转i个位置的置换,循环数为gcd(n,i)
- 翻转:
  - n为偶数时,
    - 对称轴不过顶点:循环数为n/2
    - 对称轴过顶点:循环数为n/2+1
  - n为奇数时,循环数为(n+1)/2

Polya定理小结

- 前面所讲的内容,仅适用于置换数目较少,着色没有其他限制的情况,是最简单的一类Polya定理的问题
- 复杂的Polya定理的问题还需要用到数论知识来加快速度,用排列组合或动态规划来辅助计数
- 不过,对于ACM竞赛来说,掌握简单的Polya定理就能够解决很多问题了

从博弈说起

- 一门古老的游戏;棋牌,彩票...
- 博弈论是二人或多人在平等的对局中各自利用对方的策略变换自己的对抗策略,达到取胜目标的理论
- 合作博弈/非合作博弈
- 静态博弈/动态博弈
- 完全信息博弈/非完全信息博弈

公平组合博弈

- 二人博弈
- 非合作博弈：博弈双方利益完全对立
- 动态博弈：两人轮流行动，后手知道先手的动作
- 完全信息博弈：双方都了解当前格局
- 非偶然性：无随机事件

公平组合博弈

- 游戏局面的状态数有限
- 对于同一局面,两个游戏者可操作的集合完全相同
- 无法进行任何操作时游戏结束,不能操作的一方为负
- 游戏总可以在有限步之内结束
- 假定:两个人都"足够聪明"

先手必胜/后手必胜

- 由于公平组合博弈必在有限步结束,且不会有随机事件,因此,在双方都采用最佳策略的前提下,任一局面不是先手必胜就是后手必胜
- 规定:先手胜为N(Next)局面,后手胜为P(Pre)局面

先手必胜/后手必胜

- 最终局面都是P局面
- 对于一个局面,若至少有一种操作使它变为P局面,则它是N局面
- 对于一个局面,无论如何操作都使它变为N局面,则它是P局面

抽象模型

- 所有的公平组合博弈都可以抽象成下列模型:
  - 有向无环图,某个点上有一枚棋子,双方轮流将棋子沿某条有向边移动,无法移动者为负
  - 其中,一个点表示一个局面,而一条有向边则表示一种可行操作

公平组合博弈问题解法

- 往往可以从终局面出发,逆向递推,求出任意局面是P局面还是N局面
- 但如果游戏是一系列子游戏的组合时,局面的数量非常庞大,上述方法就会变得十分低效
- 下面将引入SG函数来解决这一类问题

mex函数

- 定义函数mex,表示取最小的不属于该集合的非负整数
- 例：
  - mex{}=0
  - mex{0,1,2,4}=3
  - mex{2,3}=0

SG函数

- 定义有向图每个顶点的SG函数值
  $g(x)=mex\{g(y)|y是x的后继\}$
- 若g(x)=0,则x为P局面
  若g(x)>0,则x为N局面
- 简略证明:
  - 终局面g(x)=0
  - 若g(x)>0,则至少有一个后继y满足g(y)=0
  - 若g(x)=0,则任何后继y都有g(y)>0

游戏的和

- 设$G_1,G_2,\ldots,G_n$是n个公平组合博弈游戏
- 定义G为$G_1,G_2,\ldots,G_n$的和：G的移动规则为任选一个子游戏G_i操作一次
- 则G的SG函数值
  $g(G)=g(G_1)\oplus g(G_2)\oplus\ldots\oplus g(G_n)$
  - 注：$\oplus$为异或符号

游戏的和

- 若g(G)=0,则游戏G后手必胜
- 若g(G)>0,则游戏G先手必胜
- 证明思路:证明下列三条即可
  - 终局面SG函数值为0
  - 若SG函数值不为0,则一定存在一个操作使得新局面的SG函数值为0
  - 若SG函数值为0,则任何操作都会使新局面的SG值大于0

SG函数小结

- 有了SG函数这一工具,我们可以把这一类复杂的游戏分解成若干个简单的子游戏,然后依次求出每个子游戏的SG函数值,最后异或起来,就可以判断原游戏先手胜还是后手胜的问题了

ACM中的矩阵

- 在ACM竞赛中,有时会碰到一些与矩阵有关的问题
  - 高斯消元
  - 矩阵分解
  - 转移矩阵,矩阵乘法
  - 施密特正交化
  - 特征值,特征向量
  - 相似,合同
  - …

ACM中的矩阵

- 在线性代数的课程中基本上都已经学过
- 这些问题虽然都有标准的一般解法,但有的解法仍停留在理论上,实际上不易用计算机来解决
- ACM竞赛不会考很难的线性代数问题
- 下面只介绍一些简单的算法

高斯消元

- 将系数和常数项写成增广矩阵的形式
- 利用初等行变换将增广矩阵转化为阶梯型矩阵
- 若某一行系数全为0而常数不为0,原方程组无解,退出;
- 若出现全零的列,则原方程组有无穷多个解,全零列对应的变量为自由变量;否则原方程组只有唯一解
- 利用初等行变换将阶梯型矩阵转化为简化阶梯型矩阵
- 从下到上依次求出每个变量的值

高斯消元

- 注意事项:
  - 在将增广矩阵转化为阶梯型矩阵的过程中,最好取该列绝对值最大的元素作为主元,因为若绝对值较小,将其化成1需要乘以一个绝对值较大系数,这样会将误差放大
  - 在有无穷多解的情况下,若要求出一个可行解,只需对每个自由变量赋任意值(一般取0,1之类的数),然后解出每个主变量即可

施密特正交化

- 给定某空间的一组基$\alpha_1,\alpha_2,\dots,\alpha_s$,求该空间的一组正交基$\beta_1,\beta_2,\dots,\beta_s$
- 施密特正交化:
  - $\beta_1=\alpha_1$
  - $\beta_2=\alpha_2-((\alpha_2,\beta_1)/(\beta_1,\beta_1))\beta_1$
  - …
  - $\beta_s=\alpha_s-((\alpha_s,\beta_1)/(\beta_1,\beta_1))\beta_1$
    $-((\alpha_s,\beta_2)/(\beta_2,\beta_2))\beta_2-\dots$
    $-((\alpha_s,\beta_{(s-1)})/(\beta_{(s-1)},\beta_{(s-1)}))\beta_{(s-1)}$

矩阵分解

- 任意矩阵A可以做下列几种形式的分解
  - A=PJ,P是可逆矩阵,J是简化阶梯型矩阵(高斯消元)
  - A=QR,Q是正交矩阵($QQ'=I$),R是上三角矩阵(施密特正交化)
  - A=PU,P是投影矩阵($P^2=P$),U是可逆矩阵
  - A=PQ,P是对称矩阵($P'=P$),Q是正交矩阵

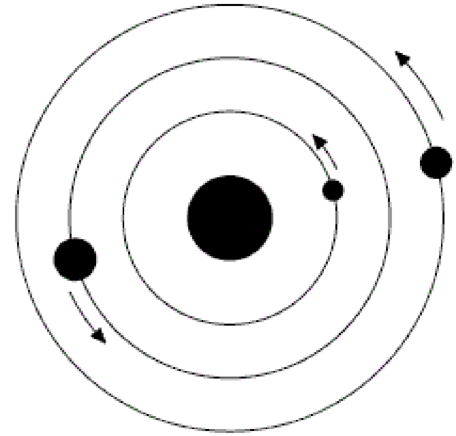
矩阵小结

- ACM竞赛中有关矩阵的问题大多数是高斯消元和矩阵乘法
- 但矩阵的一些基本问题的解法还是应该有所了解,这些有利于熟练地利用矩阵理论来思考问题,解决问题

总结

- 要解决ACM中的数学问题,关键还是要有扎实的数学基础和灵活的数学思维
- 但ACM竞赛并不是数学竞赛,一味地钻研理论是没有任何意义的,更重要的是如何利用数学知识来解决实际生活中的问题
- ACM中的数学问题涵盖了很多内容,因此,数学的各个方向都需要有一定的了解

Astronomy



- POJ3101
- 题目大意:
  - 有n个行星,它们的轨道是同一平面上的同向同心圆,且它们始终做匀速圆周运动,周期t_i已知
  - 所有卫星都处于过圆心的某条直线上的现象,称为卫星平行
  - 求相邻两次卫星平行现象的间隔时间,用分数表示

Astronomy

- 算法思路:
  - 所有卫星平行<=>任意两个卫星平行<=>相邻两个卫星平行(卫星平行具有传递性)
  - 两个卫星i,j平行的时间间隔为$|0.5/(1/t_i-1/t_j)|$(追及问题,注意是相差半周,而不是相差整周)
  - 写出相邻两个卫星平行的时间间隔$d_i=a_i/b_i$,则问题转化为求这n-1个分数的"最小公倍数"
  - 分母$p=gcd(a_1,a_2,\ldots,a_{(n-1)})$,分子$q=lcm(b_1,b_2,\ldots,b_{(n-1)})$,约分即得最终答案

The Balance

- POJ2142
- 题目大意:
  - 现有质量为a和b的砝码,数量不限
  - 要求在天平上称出质量为d的物品,天平左右均可放砝码
  - 求一种可行方案,要求:放置砝码数量尽可能少;数量相同时,总质量尽可能少

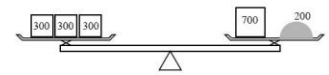


Figure 1: To measure 200mg of aspirin using three 300mg weights and one 700mg weight

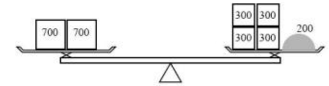


Figure 2: To measure 200mg of aspirin using four 300mg weights and two 700mg weights

The Balance

- 算法思路:
  - 问题转化:求ax+by=d的一组整数解(x,y),要求|x|+|y|尽可能小,若相等,则a|x|+b|y|尽可能小(x<0,表示砝码和物体放在同一侧)
  - 先求出不定方程的一组特解(x_0,y_0),令d=gcd(a,b),a'=a/d,b'=b/d,则通解为$x=x_0+a't$,$y=y_0-b't$(a',b'>0,t为整数)
  - 不妨设a'>b',可以证明,当|x|+|y|最小时,一定会有-b'<y<b'(用反证法证明)
  - 有了这个结论,我们只需尝试至多两个解即可

Strange Way to Express Integers

- POJ2891
- 题目大意：
  - 给定k个数对(a_i,r_i),求一个最小的正整数m,满足$m \equiv r_i \pmod{a_i}$

Strange Way to Express Integers

- 算法思路:
  - 乍一看是经典的中国剩余定理
  - 不过,仔细一想发现,题目中并没有保证a_i两两互素,这是不满足中国剩余定理的条件的
  - 好在只要令$a'_i = a_i / gcd(a_1 * a_2 * ... * a_{(i-1)}, a_i)$,下面就是直接套用中国剩余定理的结论了

Prime Distance

- POJ2689
- 题目大意：
  - 给定区间[L,U],L和U可以很大,但区间长度不超过10^6
  - 求这个区间中最近和最远的两对素数

Prime Distance

- 算法思路:
  - 直接试除?耗费时间太长
  - 直接筛法?耗费空间太大
  - Miller-Rabin素数测试?有点大材小用了吧
  - 区间长度不大->筛法+试除
  - 先用筛法求出不大于sqrt(U)的所有素数,然后用这些素数一一试除

Farey Sequence

- POJ2478
- 题目大意:
  - 求所有分母不大于n的既约真分数个数

Farey Sequence

- 算法思路:
  - 分母为x的既约真分数有$\varphi(x)$个
  - 分母不大于n的既约真分数个数为$\varphi(1)+\varphi(2)+\ldots+\varphi(n)$
  - 还记得递推式吗?
  - 递推式:质数p满足$p|x$,若$p^2|x$,则$\varphi(x)=\varphi(x/p)*a$,否则$\varphi(x)=\varphi(x/p)*(a-1)$

The Luckiest number

- POJ3696
- 题目大意:
  - 定义:只含有数字8的数为幸运数
  - 给定正整数L,求L的所有倍数中最小的幸运数的位数

The Luckiest number

- 算法思路:
  - 设最终答案为x,则x满足$(10^x-1)*8/9\equiv 0 \pmod{L}$
  - 化简上述式子:
    $\Rightarrow (10^x-1)*8\equiv 0 \pmod{9L} \Rightarrow 10^x-1\equiv 0 \pmod{9L/gcd(9L,8)}$
    $\Rightarrow 10^x\equiv 1 \pmod{9L/gcd(9L,8)}$
  - 令$m=gcd(9L,8)$,若$gcd(10,m)>1$,显然无解
  - 若$gcd(10,m)=1$,由欧拉定理:$10^{\varphi(m)}\equiv 1 \pmod{m}$
  - 不过,我们要求的是最小解,而$\varphi(m)$只是一个可行解
  - 可以证明,最小解一定是$\varphi(m)$的一个约数
  - 将$\varphi(m)$分解质因数后枚举所有约数即可

GCD & LCM Inverse

- POJ2429
- 题目大意：
  - 已知两个正整数的最大公约数和最小公倍数，求这两个数
  - 若有多解输出和最小的一组

GCD & LCM Inverse

- 算法思路:
  - 令x=lcm/gcd
  - 将X分解质因数->Pollard rho方法
  - 枚举将X分解为两个互素的数相乘

Cow Sorting

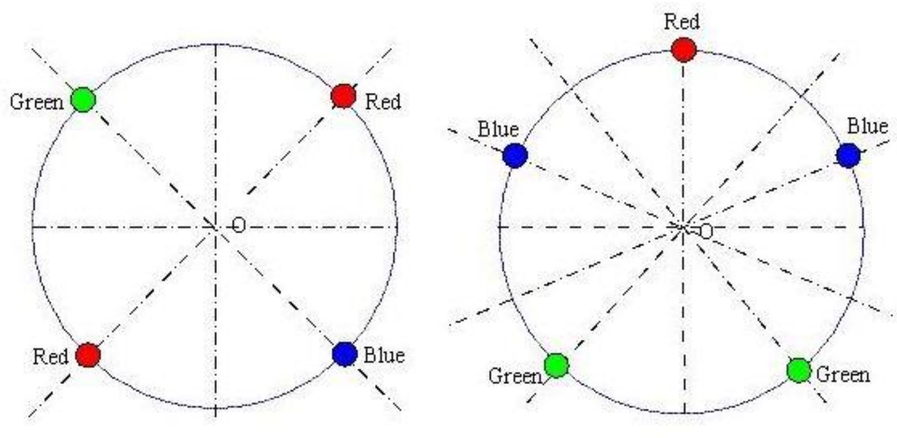
- POJ3270
- 题目大意:
  - n个两两不同的数排成一列,要求将这些数按升序排列
  - 操作:交换两个数x,y的位置,需要x+y的费用
  - 求需要的最小费用

Cow Sorting

- 算法思路:
  - 联想到循环(置换环)
  - 不同的循环相互间是没有关系的,可以分别处理
  - 一个循环,有两种处理方法:
    - 用这个循环中最小的元素,依次与相应元素交换,直到该循环内所有元素归位
    - 用这个循环中最小的元素与所有数中最小的元素交换,然后用所有数中最小的元素依次与相应元素交换,直到该循环内所有元素归位
  - 可以证明,最优的方案一定是上述两者之一

Necklace of Beads

- POJ1286
- 题目大意:
  - 将三种不同颜色的珠子串成有n个珠子的项链,旋转/翻转后相同的算同一种,求方案数



The form with n=4 The form with n=5

Necklace of Beads

- 算法思路:
  - 直接套用Polya定理的结论即可
  - 计算置换的循环数的快速方法:
    - 旋转:n个点顺时针(或逆时针)旋转i个位置的置换,循环数为gcd(n,i)
    - 翻转:
      - n为偶数时,
        - 对称轴不过顶点:循环数为n/2
        - 对称轴过顶点:循环数为n/2+1
      - n为奇数时,循环数为(n+1)/2

S-Nim

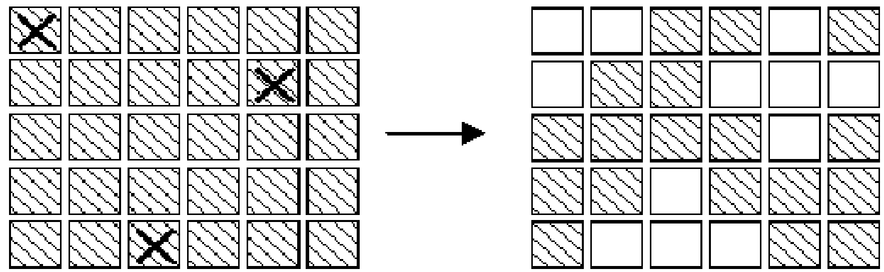
- POJ2960
- 题目大意:
  - n堆石子,已知每堆石子的数量
  - 有一个集合S,S中有k个正整数
  - 两个人轮流从这些石子堆中取石子,规则:
    - 每次只能从一堆里取
    - 取出石子的数量值是S中的一个元素
    - 不能按规则取石子者负
  - 求先手必胜还是必败

S-Nim

- 算法思路:
  - 若只有一堆石子->SG函数值
    - 根据集合S构造有向无环图
  - 多堆石子,任意两堆石子相互独立->游戏的和
    - 只要求出每堆石子的SG函数值,异或起来即为整个游戏的SG值

EXTENDED LIGHTS OUT

- POJ1222
- 题目大意:



  - 5*6的灯阵,已知初始时每盏灯的开关状态
  - 每盏灯后面有一个按钮,按下按钮后,相应的灯及其上下左右的灯的开关状态都会改变
  - 求关掉所有灯的一种可行方案

EXTENDED LIGHTS OUT

- 算法思路:
  - 把每个按钮的使用次数设为未知数,注意到一个按钮使用两次等于没用,因此使用次数只能是0或1
  - 对于每盏灯,都可以列出一个xor方程
  - 接下来就是利用高斯消元来解这个xor方程组了

High-Dimensional Vector Correspondence

- POJ3707
- 题目大意：
  - 给定两个m维向量组p1,p2,...,pn和q1,q2,...,qn,求能否经过一系列的翻转和旋转操作,使得p1,p2,...,pn变成q1,q2,...,qn

High-Dimensional Vector Correspondence

- 算法思路:
  - 仔细分析题目可以看出,本题实际上是给定两个矩阵P和Q,要求一个正交矩阵H,使得HP=Q
  - 利用施密特正交化,将P分解P=TU,其中T为正交矩阵,U为上三角矩阵(注意P的秩可能小于n)
  - 令A=HT,则AU=Q,转置得U'A'=Q',此时U'为下三角矩阵
  - 用类似高斯消元的办法求出A'(注意无解和无穷多解的情况)
  - 通过逆矩阵的一些相关知识即可求出H

- 今天的内容就到这里
- 谢谢