【3】扩展欧几里得/费马小定理/欧拉定理/逆元/筛法/卢卡斯定理

一次同余方程$ax = b \pmod p$，有解条件：$gcd(a,p)|b$

设a,b,p约去$gcd(a,p)$为a',b',p'，则：$a'x = b' \pmod{p'}, a' \perp p'$

故得到特解：$x_0 = a'^{-1}b' \pmod{p'}$，显然通解$x \pmod p$有$gcd(a,p)$种

【8】群论基础与$Polya$计数法定理，$bunside$引理

- 群作用
  - 定义
    - **G 为一个群，X 为一个集合，那么 G 在 X 上的作用定义为一个映射：**
      $G \times X \to X$，**其将一个有序对 (g,x) 映射到 $g[x]$，其中 $g \in G, x \in X, g[x] \in X$**
  - 性质
    - $e[x] = x$
    - $\forall g_1, g_2 \in G, g_1[g_2[x]] = (g_1g_2)[x]$
  - Cayley定理推论
    - 任意一个群在集合上的作用的实质都是与其同构的一个变换群在集合上的作用
  - 群作用下可相互转化的等价关系：
    - $\forall x, y \in X, x \sim y$ **当且仅当** $\exists g \in G,\ s.t.\ g[x] = y$
  - 轨道
    - 定义
      - O_x **为元素** x **所在的由关系** $\sim$ **导出的等价类，被称为** x **所在的轨道**
    - 性质
      - $\forall x, y \in X, x \sim y$ **当且仅当** $O_x = O_y$
      - **任意两个轨道要么相同要么不交**
      - **所有不同轨道给出了集合** X **的一个划分，即在** X **每条不同轨道上选取一个元素组成集合** I **的话，有**
        $X = \bigcup_{x \in I} O_x$
      - $\sum_{x \in X} \frac{1}{|O_x|} = n$ 为不同轨道个数
  - 稳定化子
    - 定义
      - **对于** $x \in X$，$G_x = \{g \in G | g[x] = x\}$ **被称为群作用下元素** x **的稳定化子**
      - **稳定化子必定是作用群** G **的子群**
  - Lagrange公式
    - x **所在轨道元素个数等于** x **的稳定化子的不同陪集数**
    - $|O_x| = |G/G_x| = \frac{|G|}{|G_x|}$
- $bunside$引理
  - 群作用应用：将X **看作是所有染色方案组成的集合，** G **看作是给定对称意义下的所有变换构成的群，轨道** O_x **包含的元素都是在对称意义下与** x **相同的染色方案，** G_x **中元素都是保持染色方案** x **不变的变换**，所需要求的在对称意义下本质不同的非同构的染色方案数即**不同轨道的数量**n
  - 不动点集合：$\psi(g) = \{x \in X | g[x] = x\}$ **为在** g **作用下不变的元素的集合**
  - 算两次思想：考虑不动元祖(g,x)表示被作用元素x为变换g下的一个不动点，对其进行计数
    - $\sum_{g \in G} |\psi(g)| = \sum_{x \in X} |G_x|$。
    - $\sum_{g \in G} |\psi(g)| = \sum_{x \in X} \frac{|G|}{|O_x|} = |G| \sum_{x \in X} \frac{1}{|O_x|} = |G|n$
    - $n = \frac{1}{|G|} \sum_{g \in G} |\psi(g)|$
    - **群中元素的不动点个数的平均值就是不同轨道的个数**
  - 简单的例题

    - 6 种颜色对立方体的六个面染色，每个面颜色必须不同，立方体可以在空间任意转动，求不同染色方案数。
- $Polya$计数法
  - 基础
    - **A 为被染色块构成的集合，C 为所有颜色构成的集合，$|A|=n,|C|=m$，那么任意一个映射 $f:A\to C$ 都代表了一个染色方案，被称为一个染色函数。C^A 即表示所有的染色函数 f 构成的集合。G 为 A 上的一个置换群，包含了染色块之间"对称关系"的含义。X 是所有染色函数组成的集合而不再是不同染色方案的集合。**
    - 定义G 作用在 C^A 上的**群作用**：$g[f]=fg^{-1}$
    - **对称意义下的不同染色方案数仍旧等于这个群作用中的不同轨道的个数**
  - 核心
    - 考虑g为一个置换，考虑其包含的一个轮换，其中涉及到的元素，必须被染成相同的颜色
    - 因此在置换g作用下保持不变的方案有：$m^{k(g)}$，$k(g)$为其轮换个数
    - **不同轨道个数为** $\dfrac{\sum_{g\in G}m^{k(g)}}{|G|}$
    - 关键在于如何求置换群中轮换个数的分布，完整置换群的轮换分布可以与生成函数结合
  - 定理
    - **已知 $|A|=n,|C|=m,G$ 为 A 上的置换群，则所有不同轨道个数为** $\dfrac{\sum_{g\in G}m^{k(g)}}{|G|}$。
- 带权$bunside$引理
  - 权函数
    - 由于同一个轨道中的染色方案是等价的，因此应该有相同的权，令$\omega(x)$表示染色方案x的权，轨道的权等于其包含元素的权
    - 我们不再求不同轨道数，转而求不同轨道权之和$\sum\limits_{i=1}^{N}\omega(O_i)$，当权取1时被化归
  - 核心
    - $\sum\limits_{i=1}^{N}\omega(O_i)=\sum\limits_{x\in X}\dfrac{\omega(x)}{|O_x|}$。
    - $\sum\limits_{i=1}^{N}\omega(O_i)=\dfrac{1}{|G|}\sum\limits_{g\in G}\sum\limits_{x\in\psi(g)}\omega(x)$。
    - **所有轨道的权之和等同于群中每个元素对应的不动点的权的平均数**
- 带权$Polya$定理
  - 权函数
    - 考虑$f\in C^A$ **为染色函数，其中 A 为被染色块集合，C 为颜色的集合，$|A|=n,|C|=m$，权函数 ω 定义在集合 C 上，G 仍为 A 上的一个置换群**
    - $\omega(f)=\prod\limits_{a\in A}\omega(f(a))$ **为染色函数 f 的权，染色函数的权等同于其将所有染色块映射到的颜色所具有的权的积**
  - 定理
    - $\omega(f)=\prod\limits_{a\in A}\omega(f(a))=\omega(c_1)^{|A_1|}\dots\omega(c_{k(g)})^{|A_{k(g)}|}$。其中$k(g)$表示$g$的轮换个数，$A_i$表示第$i$个轮换的符号集合，即涉及到的被作用元素集合；$c_i$表示第$i$个轮换被染的颜色。
    - $\sum\limits_{f\in\psi(g)}\omega(f)=\sum\limits_{c_1,\dots,c_{k(g)}\in C}\omega(c_1)^{|A_1|}\dots\omega(c_{k(g)})^{|A_{k(g)}|}$。

      $=\prod\limits_{i=1}^{k}(\sum\limits_{c\in C}\omega(c)^i)^{c_i(g)}$。其中$c_i(g)$表示$g$中长度为$i$的轮换个数。$k=|A|$，即最大轮换长度。
    - 由带权的Burnside定理，$\sum\limits_{i=1}^{N}\omega(O_i)=\dfrac{1}{|G|}\sum\limits_{g\in G}\sum\limits_{f\in\psi(g)}\omega(f)=\dfrac{1}{|G|}\sum\limits_{g\in G}\prod\limits_{i=1}^{k}(\sum\limits_{c\in C}\omega(c)^i)^{c_i(g)}$

若记$\omega_i=\sum_{c\in C}[\omega(c)]^i$，表示一个长度为$i$的轮换的染色权之和，若权为1，即退化为方案数。

则得到：$\sum_{i=1}^{N}\omega(O_i)=\frac{1}{|G|}\sum_{g\in G}(\omega_1^{c_1(g)}\dots\omega_n^{c_n(g)})$

- 生成函数表示与优化
  - 令$\omega(c)$表示颜色c对应的占位符x_c，而不是简单的数值权，带入$Polya$定理可以很容易得到非同构的一些方案数，其包含的细节更加丰富，很容易和生成函数结合起来：例如表示每种颜色各使用多少个的非同构方案数等。
  - 令$\omega_i=\sum_{c\in C}x_c^i$表示轮换占位符，$\prod_{i=1}^{n}x_i^{c_i}$表示种类占位符，排列的$Cycle\ Indicator/Index$即是关于种类占位符的，只需要带入$x_i=\omega_i$即得到关于颜色占位符$\prod_{c\in C}x_c^{cnt_c}$的生成函数，此项前面的系数即表示非同构的颜色$c$恰好使用了$cnt_c$次的方案数(当然要除以$|G|$)
- 简单例题
  - 求n个点的**边非同构**无向图(点不带标号)，求出恰好包含i条边的答案ret_i
    - 染色对象集合为$A=E$即完全图边集，颜色集合$C=\{x,y\}$表示两种颜色，取或者不去(y可令为1)，置换群为全变换群$G=S_{|A|}$，按照上面的思路即可得到n个点i条边的答案
    - $n=3$的情况：
      $$\sum_{i=1}^{N}\omega(O_i)=\frac{(b+w)^3+3(b+w)(b^2+w^2)+2(b^3+w^3)}{6}=b^3+w^3+b^2w+bw^2$$
      表示共有4种，每种情况各1种
    - 注意由于占位符个数太多，会产生维数爆炸的情况，不便于求大的数据规模，因此当$i\le k$时，我们没有必要考虑$\omega_i,i>k$，直接令其为1即可；比如说题目只要求$i\le 4$
    - 就此问题，每种必然答案为1，总方案答案为$|E|+1$
  - 求n个点的**边非同构**无向图(点不带标号)，ans_n
    - 由于不要求具体的颜色数量限制，因此令权值为1即可，即令轮换占位符中的$x_c=1$
    - 令$M=\binom{n}{2}$，由定理知，不同的轨道数$N=\frac{1}{|G|}\sum_{g\in G}2^{k(g)}=\frac{1}{M!}\sum_{i=1}^{M}2^iS(M,i)=M+1$，其中$S$表示第一类斯特林数
    - 最后一步用到了斯特林数的行生成函数，这是一个上升阶乘幂

【9】离散对数与原根

在线求原根 维基百科：原根

- 关于原根存在的充要条件：
  - 原根存在，则$m=1,2,4,p^a,2p^a$，其中p为奇质数且$a\ge 1$
  - $m\ge 3$原根存在，$x^2=1\pmod p$存在两解(模p意义下)
  - $m\ge 3$，$[1,m]$中与m互质的数的乘积模m为-1有原根，为1则无原根
    $$\prod_{\substack{k=1\\ged(k,m)=1}}^{m}k\equiv\begin{cases}-1(\bmod m),m\in[1,2,4,p^e,2p^e],p\text{为奇质数}\\1(\bmod m),\text{否则}\end{cases}$$
    上述结论在模m^2意义下未必成立，若成立则m为$Wilson\ numbers$

    上述结论可推广到***任意有限交换群***，群中所有元素的乘积等于单位元或者一个阶为2的元素维基百科

    关于威尔逊商数，其同余递推式与伯努利数有关
    $$W(p)\equiv B_{2(p-1)}-B_{p-1}\pmod p$$
    $$p-1+ptW(p)\equiv pB_{t(p-1)}\pmod{p^2}$$
- 求原根算法$O(glog^2p)$
- $BSGS$及其扩展求离散对数$a^x=b\pmod p$即$x=log_ab\pmod{\phi(b)}$,p未必与a互质

  其算法实质是分块思想，扩展时两边同时消因子，复杂度$O(\sqrt{p})$
- 解高次同余方程$x^a=b\pmod p$，p为质数
  $$log_g^x=log_{g^a}x^a=log_{g^a}b\pmod{\phi(p)}$$
  $$alog_g^x=log_g^b\pmod{\phi(p)}$$
  转化为第二种情形，即$(g^a)^x=b\pmod p$

  如果要求所有$[0,p)$中的解，最好转化后用$exgcd$解普通同余方程或者求逆元，得到一个最小特解$log_g^x=a'^{-1}(log_g^b)'=y_0\pmod{\phi(p)'}$，则通解显然为：$y_0+k\phi(p)',k\in[some\ range]$

  通解在$\pmod{\phi(p)}$意义下有$gcd(a,\phi(p))$种，这与原高次同余方程的解$\pmod p$显然一一对应
- 解高次同余方程$x^a=b\pmod p$，$p\in Z^+$
  - 特判$p=1$，特判$a=0$

- 将p分解质因子，要求2的指数不超过2，否则将因为没有原根，而报无解
- 对每个p^k单独考虑，化为模数为质数的幂次的情况，最后中国剩余定理crt合并
- 将b规范化($b \in [0,p^k)$)，然后分解为$b=b'p^{cnt}$，其中$p^{cnt}||b$
  - $cnt \geq k$，即$p^k|b$，此时b必为0，特判之
  - $cnt>0$，即$gcd(b,p^k)>1$，方程化为：$x^a/p^{cnt} = \frac{b}{p^{cnt}} \pmod{p^{k-cnt}}$
    - 特判$a \not| cnt$，此种情况认为无解
    - 化为：$(\frac{x}{p^{cnt/a}})^a = \frac{b}{p^{cnt}} \pmod{p^{k-cnt}}$，转化为互质情形
  - $cnt=0$，即互质情形，与上一个知识点类似，即当p为质数时
- 常常构造最小非负整数解(crt直接合并)，求解$\in [0,p)$个数，求所有解等
- 求解个数，可以根据crt证明解的一一对应，即简单的利用乘法原理即可
- 模板支持：$a \geq 0, 1 \leq p < 10^{12}$，分解后每类均有原根

- 原根的性质：
  - 数量：$\phi(\phi(p))$
  - 可相互表示：若g为原根，则g^k亦为原根，其中$k \perp \phi(p)$
  - 若g是质数p的一个原根，则p^k原根为：
    - $g+p$，当$g^{p-1}=1 \pmod{p^2}$
    - g，否则
  - 若g是p^k的一个原根，则$2p^k$原根为：
    - g和$g+p^k$其中任意一个奇数
  - 若质数$p!=3$，则其所有原根之积$ret=1 \pmod{p}$
  - $\forall$质数p，其所有原根之和$ret=\mu(p-1) \pmod{p}$
- 注意点：
  - 不确定是否不同余0，就要讨论及特判
  - 注意在数学上$0^0=1$，故同余方程$a^x=b \pmod{p}$，当$a=b=0$时，注意最小非负整数解$x_0=[p==1] \oplus 1$

【33】$Kummer$定理

- $Legendre's\ formula$
  - $\nu_p(n!)=\sum_{i=1}^{\infty}\left\lfloor \frac{n}{p^i} \right\rfloor = \frac{n-s_p(n)}{p-1}$
  - 其中$\nu_p(x)$表示x在p进制下末尾0个数
- $Multinomial\ coefficient\ generalization\ for\ Kummer$
  - $\nu_p\left(\binom{n}{m_1,\ldots,m_k}\right)=\frac{1}{p-1}\left(\sum_{i=1}^{k}S_p\left(m_i\right)-S_p(n)\right)$
  - 其中$S_p(x)$表示x在p进制下的数位之和

【58】代数：$(min,+)$卷积/最值反演/$min-max$容斥

注意为叙述方便，以下公式中，$\subset$类似符号常常表示非空包含；

对$\emptyset$不定义$min/max/gcd/lcm$等函数；

且$d|S$表示$\forall x \in S, d|x$，此处$S \neq \emptyset$.

- 基础公式

$$\max(S)=\sum_{T\subseteq S}(-1)^{|T|+1}\min(T)$$

$$\min(S)=\sum_{T\subseteq S}(-1)^{|T|+1}\max(T)$$

- k大值扩展

$$kth\max(S)=\sum_{T\subseteq S}(-1)^{|T|-k}\binom{|T|-1}{k-1}\min(T)$$

主要用于一些求期望的题目，求一些随机变量的最大值可以转化为求这些随机变量最小值实现

$kthmax$容斥这个系数是可以关于k进行递推的，如洛谷4707

- lcm与gcd扩展

$$lcm(S)=\prod_{T\subset S}gcd(T)^{(-1)^{|T|+1}}$$

- 一个加速求值的应用

如果函数f已知，且满足gcd分配率，即：

$$f(gcd(S))=gcd(f(S))$$

我们要求：

$$g(n)=\prod_{T\subset[n]}f(gcd(T))^{(-1)^{|T|+1}},[n]=\{1,\cdots,n\}$$

若函数f是某个函数h的偏序前缀积：

$$f(n)=\prod_{d|n}h(d)$$

则有

$$g(n)=\prod_{d=1}^{n}h(d)$$

证明过程：

令$S=[n]$，令$p=\lfloor\frac{n}{d}\rfloor$.

将$f(n)$带入到$g(n)$表达式中得：

$$\begin{aligned}g(n)&=\prod_{T\subset S}\prod_{d|gcd(T)}h(d)^{(-1)^{|T|+1}}\\&=\prod_{T\subset S}\prod_{d|T}h(d)^{(-1)^{|T|+1}}\\&=\prod_{d=1}^{n}h(d)^{\sum_{T\subset S,d|T}(-1)^{|T|+1}}\end{aligned}$$

$$\text{其中}\sum_{T\subset S,d|T}(-1)^{|T|+1}=\sum_{k=1}^{p}\binom{p}{k}(-1)^{k+1}=1,\text{命题得证。}$$

- 应用扩展
  - 假设S为一般有穷正整数集合，则令$S_d\subset S$表示$\{x|x\in S,d|x\}$，$S_d\neq\emptyset$.令$G=\{x|x|y,y\in S\}$为S的约数集，G显然满足一种偏序单调性.

    $$\sum_{T\subset S_d}(-1)^{|T|+1}=1$$

    $$g(S)=\prod_{T\subset S}f(gcd(T))^{(-1)^{|T|+1}}=\prod_{d\in G}h(d)^{\sum_{T\subset S_d}(-1)^{|T|+1}}=\prod_{d\in G}h(d)$$

    也就是说，我们推广了函数g在更一般的有限集S中与函数h的关系，简单的类比可以得到：

    $$f(n)=\sum_{d|n}h(d)\quad\Longrightarrow\quad g(S)=\sum_{d\in G}h(d)$$

    $$\text{其中：}\ g(S)=\sum_{T\subset S}(-1)^{|T|+1}f(gcd(T))$$
  - 令$f(S)$表示$\{f(x)|x\in S\}$，注意$g(S)$不是这样定义的，则

    $$\begin{aligned}lcm(f(S))&=\prod_{T\subset f(S)}gcd(T)^{(-1)^{|T|+1}}=\prod_{T\subset S}gcd(f(T))^{(-1)^{|T|+1}}\\&=\prod_{T\subset S}f(gcd(T))^{(-1)^{|T|+1}}=g(S)=\prod_{d\in G}h(d)\end{aligned}$$

    推论：当$S=[n]$时，$lcm(f(S))=\prod_{d=1}^{n}h(d)$.

    仍然可以做简单的类比，得到$gcd(f(S))$.
  - 与类$lucas$数列结合，求$lcm(f(S))$，因为该数列f满足gcd分配率，处理出f之后，莫比乌斯反演得到h，利用集合S构造下标$bool$数组S，对其做莫比乌斯变换，得到约数偏序集G，求G中所有元素h函数值之积

【59】几何：$Szemer\acute{e}di-Trotter\ theorem$

定理内容：

It asserts that given n points and m lines in the Euclidean plane, the number of incidences (i.e., the number of point-line pairs, such that the point lies on the line) is

$$O(n^{2/3} * m^{2/3} + n + m)$$

【60】几何：三维叉积的应用

求过两点的直线方程一般形式，判断两直线关系，并且求交点坐标

代码短，精度高，不容易出错

参考：https://blog.csdn.net/abcjennifer/article/details/7584628

摘要如下：

一般方程法：

直线的一般方程为$F(x) = ax + by + c = 0$。既然我们已经知道直线的两个点，

假设为$(x_0, y_0), (x_1, y_1)$，那么可以得到$a = y_0 - y_1, b = x_1 - x_0, c = x_0 y_1 - x_1 y_0$。

因此我们可以将两条直线分别表示为

$$F_0(x) = a_0 x + b_0 y + c_0 = 0, F_1(x) = a_1 x + b_1 y + c_1 = 0$$

那么两条直线的交点应该满足

$$a_0 x + b_0 y + c_0 = a_1 x + b_1 y + c_1$$

由此可推出

$$x = (b_0 c_1 - b_1 c_0)/D$$

$$y = (a_1 c_0 - a_0 c_1)/D$$

$$D = a_0 b_1 - a_1 b_0,\ (D为0时，表示两直线平行)$$

【66】组合：一个经典的组合问题转化结论

对于前n个正整数构成的集合，求所有k元子集元素乘积之和（$1..n$中k乘积之和）

答案：$ans = S[n+1][n-k+1]$,S表示第一类斯特林数

【80】组合：gcd卷积与lcm卷积

卷积是一种比较耗费计算资源的基础数学运算，在数学上我们处理卷积的一般方法都是通过各种变换或者转化，使得问题转化为比较容易计算的计算类型(比如序列点积)，想方设法加速计算的效率，当然这需要卷积本身具有一定的数学性质。

***gcd*卷积**是说我们需要计算的卷积公式为：

$$c_k = \sum_{(i,j)=k} a_i * b_j$$

令A, B, C分别为a, b, c的类点值序列(闭区间$[1, n]$,下标超过n值为0)，

其中一个定义如下：

$$A_k = \sum_{k|d} a_d$$

那么即有：

$$C_k = \sum_{k|d} c_d = \sum_{k|d} \sum_{(i,j)=d} a_i * b_j = \sum_{k|(i,j)} a_i * b_j$$
$$= \sum_{k|i} \sum_{k|j} a_i * b_j = (\sum_{k|i} a_i) * (\sum_{k|i} b_i) = A_k * B_k$$

我们得到：

$$C = A \bullet B$$

序列反演公式：

$$a_k = \sum_{k|d} A_d * \mu_{\frac{d}{k}}$$

***lcm*卷积**是说我们需要计算的卷积公式为：

$$c_k = \sum_{[i,j]=k} a_i * b_j$$

令A, B, C分别为a, b, c的类点值序列，其中一个定义如下：

$$A_k=\sum_{d|k}a_d\Leftrightarrow A=a\circ I$$

那么即有：

$$C_k=\sum_{d|k}c_d=\sum_{d|k}\sum_{[i,j]=d}a_i*b_j=\sum_{[i,j]|k}a_i*b_j$$
$$=\sum_{i|k}\sum_{j|k}a_i*b_j=(\sum_{i|k}a_i)*(\sum_{i|k}b_i)=A_k*B_k$$

我们得到：

$$C=A\bullet B$$

序列反演公式：

$$a_k=\sum_{d|k}A_d*\mu_{\frac{k}{d}}=\sum_{x*y=k}A_x*\mu_y\Leftrightarrow a=A\circ\mu$$

更多信息可以参见模板和配套的md说明文件

【85】组合：k进制fwt/高维广义快速离散傅里叶变换fft

本质：有穷高维循环卷积fft，循环周期为k

http://www.cnblogs.com/TinyWong/p/10351109.html

https://www.cnblogs.com/reverymoon/p/10197711.html

需要变换的下标每个数都是n位的k进制数，那么下标空间为$[0,k^n)$

这相当于一种n维的($n=log_kN$)每维长度为k的fft循环异或卷积变换(k进制不进位加法)，按照高维fft的思路，应该按照某种维度的排列顺序(比如从高位到低位)，逐维正变换，然后每个位置独立做点积，最后再逐维逆变换(对应一种递归或者迭代过程)

2进制宏观上正变换的实质：$c_i=\sum_{j=0}^{2^k-1}(-1)^{num(j\&k)}a_j,i\in[0,2^k-1]$

k进制微观上每维正变换的实质，与fft相似，是**向量左乘一个变换矩阵**T，满足$T(i,j)*T(i,k)=T(i,j\oplus k)$

我们采用k阶单位根的***范德蒙德矩阵***:

$$T=\begin{bmatrix}1&1&1&\dots&1\\1&w_k^1&w_k^2&\dots&w_k^{k-1}\\1&w_k^2&w_k^4&\dots&w_k^{2(k-1)}\\\dots&\dots&\dots&\dots&\dots\\1&w_k^{k-1}&w_k^{2(k-1)}&\dots&w_k^{(k-1)(k-1)}\end{bmatrix}$$

其行列式为：

$$det(T)=\prod_{i<j}(x_i-x_j),\ x_i\neq x_j$$
$$det(T)\neq0\leftrightarrow T_{\text{可逆}}$$

其逆矩阵为：

$$T^{-1}=\frac{1}{k}\begin{bmatrix}1&1&1&\dots&1\\1&w_k^{-1}&w_k^{-2}&\dots&w_k^{-(k-1)}\\1&w_k^{-2}&w_k^{-4}&\dots&w_k^{-2(k-1)}\\\dots&\dots&\dots&\dots&\dots\\1&w_k^{-(k-1)}&w_k^{-2(k-1)}&\dots&w_k^{-(k-1)(k-1)}\end{bmatrix}$$

证明用到结论：

$$[n|t]=\sum_{i=0}^{n-1}w_n^{ti}$$

注：2进制下的特殊情形

$$T=\begin{bmatrix}1&1\\1&-1\end{bmatrix}=\begin{bmatrix}0.5&0.5\\0.5&-0.5\end{bmatrix}^{-1}$$

一种**实现思路**是：

采用递归，传入参数FWT(a,S,n,op)，表示的是将a数组中以S开头的长度为n的一段区间进行正/逆变换(op=1为正/op=-1为逆)，那么首先将这段区间平均分为k段，每段长度n/k，逐段递归FWT，然后回溯合并时扫描段长(穷举一个段中偏移为i的位置)，将这些位置取出来构成一个k维向量，将这个向量左乘单位根矩阵T或者T^{-1}，变换后按顺序放置回去，对每个偏移量i都这样操作即可

数的表示技巧：单位根表示法

注意到一个数x，需要与k阶单位根做乘积，正常的思路是用一个$pair<double,double>$保存其复数的实部和虚部，用复数的乘法规则运算，但是这样显然精度损失严重，最终不容易回到其整数形态

注意到数x一定具有形式：$x=\sum_{i=0}^{k-1}a_i\omega^i$，故可以使用$k$维向量$\vec{a}$表示数$x$

那么普通的加减就是模意义下逐位加减，普通的乘法变为k维向量的**循环卷积**即可

而与ω^t相乘相等于，此向量循环右移t位

考虑最后如何由向量$\vec{a}$得到整数值x:

$$令cnt=k\&-k,\ L=k/cnt$$

$$容易发现，[0,k)这些下标可以分成cnt组，每组L个元素，分组的依据是一个下标i\&cnt的值的不同$$

$$比如k=20可以分为以下4组：$$

组/元素	/4=0	/4=1	/4=2	/4=3	/4=4
组0	0	4	8	12	16
组1	1	5	9	13	17
组2	2	6	10	14	18
组3	3	7	11	15	19

当k为偶数时，0和$k/2$这两个位置保存整数值；当k为奇数时，只有0这个唯一的位置保存整数值

我们每组独立考虑，首先0和$k/2$不会分到同一组，因为$k/2$存在，则必定k为偶数，k为偶数，则可以先按照每个下标的最后一位是0或者1暂时分成两组，若0和$k/2$同分在组A，则x轴上方和下方显然有一样多的元素属于组A，那么组A显然包含了偶数个元素，则组A，还可以继续拆分为两组，重复应用这个过程，由无穷递降法知矛盾

基本事实：每组单位根之和为0，这是因为：

$$x^n-1=\prod_{i=0}^{n-1}(x-\omega^i)=(x-1)(x^{n-1}+x^{n-2}+\cdots+1)$$

$$约去x-1得到：$$

$$x^{n-1}+x^{n-2}+\cdots+1=\prod_{i=1}^{n-1}(x-\omega^i)$$

$$而\sum_{i=0}^{L-1}\omega_k^{cnt*i+t}=\omega^t(\sum_{i=0}^{L-1}\omega_{cnt*L}^{cnt*i})=\omega^t(\sum_{i=0}^{L-1}\omega_L^i)=0$$

若一个组不含0和$k/2$，那么这个组元素都相同，可消去为0；若含有其中一个，则其余元素全相同，可消除为除0下标之外元素为0；那么我们最多考虑两个组贡献，就是$a[t]-a[t+cnt],t\in\{0,k/2\}$

可以得到模板代码求整数值如下：

```
inline ll value() {
    int cnt=K&-K,L=K/cnt; ll ret=a[0]-(L>1)*a[cnt];
    if (K&1^1) ret-=a[K>>1]-(L>1)*a[(K>>1)+cnt],ret%=_p;
    return ret;
}
```

单位根表示法的优化：

很显然，对于一个偶数进制k(k为奇数的两倍)，除了用k阶单位根表示一个数，还可以用k/2阶单位根，这主要是因为有类似于折半引理的复数性质，可以节约一些常数；如果这里可以找到一些更本质的性质，也许可以推广下去，或者针对具体的进制，具体的分析

数的表示技巧：模域表示法

这是一种可以让高进制卷积，加速好几倍同时没有精度误差的方法

- 适用条件：只适用于**特殊进制**k和**模数**p，要求p为质数且$k|p-1$.

    - 模p意义下循环群阶为$p-1$，元素阶以及循环子群的阶均为其约数
    - 主要思想是用模域中的数刻画有理数，实数，以及复数域中的数
    - $\sqrt[m]{t}$用m次剩余去刻画，虚数单位i使用$\sqrt{-1}$刻画，一定条件下，复数可以得到刻画
    - 令g为模p原根，k阶单位根，共k个，构成循环子群；原根g为其生成元，$w_k=g^{\frac{p-1}{k}}$
    - 模数为10^9+9或998244353最为适用，进制k为$2,3,4,7,8$最为得当
- 表示方法：根据前面的分析，一个数$x=\sum_{i=0}^{k-1}a_iw_k^i$，带入单位根即得到模域表示
- 基本运算
    - 乘法转化为循环卷积，这里即普通乘法，衍生出乘方
    - 左右移，通过乘除单位根的幂次得到
    - 实数求值，该值就是答案，因为虚部为0，在模域里也抵消了
    - 逆向分解，不大可能，因为相当于降维了，信息与高维比不充分
- 主要用途
    - 对卷积运算加速常数
    - 在复数域中扩展多项式操作，比如生成函数带入单位根，或者单位根反演，支持三角函数操作等
    - ntt本身就是一种特殊的2^k阶单位根处的多点求值

求逆技巧：

注意在FWT的逆变换中，最中间的赋值语句要除以一个进制数，或者最终结果每个元素要除以一个长度$N=k^p$

而FWT正变换中没有除法，所以最终结果每个元素只要除以N^t，其中t为逆变换的总次数

模数变更：

当模数不是一个质数的时候，考虑进制数k可能和模数不互质，那么将k分解为$k=k'*q$，其中q使用数学技巧转化掉(事实上要视情况而定，可能不能转化掉)，而k'是互质成分，存在逆元

当存在特殊模数的时候，比如2^{58}，可以不取模，常数可能会小很多

公共性质：

卷积定理表明：

$$T(a\oplus b)=T(a)\cdot T(b)$$
$$T(a+b)=T(a)+T(b)$$
$$T(h(a,b))=h(T(a),T(b))$$
$$T(h(\overrightarrow{a}))=h(\overrightarrow{T}(\overrightarrow{a}))$$

其中$h(a,b,c)$是关于多项式a,b,c之间的组合多项式函数

【91】代数：暴力多项式取模m^2logn，用于特征多项式法解k阶常系数线性齐次递推数列(k>=1000)

蕴含着：Cayley–Hamilton theorem

https://en.wikipedia.org/wiki/Cayley%E2%80%93Hamilton_theorem

参考资料：

https://blog.csdn.net/joker_69/article/details/80869814

https://blog.csdn.net/hzj1054689699/article/details/85683342

有模板题和模板：***LinearRecursion***ZP版本$O(m^2logn)$

https://www.lydsy.com/JudgeOnline/problem.php?id=4161

简要笔记：

$O(m^2logn)$可以使用长除法，模拟多项式除法/取模(被除$deg=n\le 1e18$,除数$deg=m\le 1000$)，参见模板

令递推变换矩阵为$A_{m\times m}$，其特征多项式$G(\lambda)$为：

$$G(\lambda)=|\lambda I-A|=\lambda^m-\sum_{i=0}^{m-1}a_{m-i}\lambda^i=[-a_m,-a_{m-1},-a_{m-2},\dots,-a_1,1]$$

根据$Cayley-Hamilton$定理，有：$G(A)=0$

若$x^n=F(x)G(x)+C(x),deg(C)<m$，则有$A^n=C(A)$

更进一步地：$A^n \overrightarrow{x} = C(A)\overrightarrow{x} = (\sum_{i=0}^{m-1} c_i A^i)\overrightarrow{x} = \sum_{i=0}^{m-1} c_i (A^i \overrightarrow{x})$

假设我们取具体而特定的：$\overrightarrow{x}^T = [x_{m-1}, x_{m-2}, x_{m-3}, \ldots, x_1, x_0]$

为了得到x_n，我们应该左乘A^{n-m+1},取结果向量第一个元素$x_n = (A^{n-m+1}\overrightarrow{x})[1]$

我们得到：

$$x_n = \sum_{i=0}^{m-1} c_i x_{m-1+i}$$

上述差卷积需要已知：$X' = [x_{m-1}, x_m, x_{m+1}, \ldots, x_{2m-3}, x_{2m-2}] = [x_i], i \in [m-1, 2m-2]$

此种方法可能还需要我们多算$O(m)$项，如果我们左乘A^n取结果向量第m个元素(最后一个)，亦即：

$$x_n = (A^n \overrightarrow{x})[m] = \sum_{i=0}^{m-1} c_i (A^i \overrightarrow{x})[m] = \sum_{i=0}^{m-1} c_i ((A^i \overrightarrow{x})[m]) = \sum_{i=0}^{m-1} c_i x_i = \overrightarrow{C} \cdot \overrightarrow{X}$$

$$\text{其中：} \overrightarrow{C} = [c_i], \overrightarrow{X} = [x_i], i \in [0, m-1], \text{另 } \cdot \text{表示向量点积}$$

可以看到，这种方法没有什么冗余计算，$\overrightarrow{C}$和$\overrightarrow{X}$均为已知的，只要直接做一次点积即可

而$\overrightarrow{C}$通过多项式x^n对$G(x)$取模得到，由于n巨大，所以不能用普通的多项式取模模板，考虑使用快速幂$Pow(x, n, G)$,令多项式x^t在模G意义下不断自乘即可，故我们需要写一个模板函数，$mul(A, B, G)$表示模G意义下多项式A和B乘积，其中如果多项式乘和取模用暴力$O(m^2)$，则此方法复杂度$O(m^2 logn)$，而如果使用多项式模板(ntt实现)，则复杂度为$O(mlogmlogn)$

如果我们只知道$\overrightarrow{a} = [a_i], i \in [1, m]$，和$x_0$，而不知道整个$\overrightarrow{X}$，现在要求$x_n$，就需要先求$\overrightarrow{X}$：

$O(m^2)$可以暴力线性递推，$O(mlog^2 m)$可以使用分治fft/ntt，$O(mlogm)$可以使用以下生成函数方法：

$$\text{令} \overrightarrow{H} = \sum_{i=0}^{+\infty} x_i z^i \text{为普通型生成函数}$$

$$\overrightarrow{H} = \frac{\overrightarrow{H}(1 - \overrightarrow{a})}{1 - \overrightarrow{a}}$$

$$\text{故：} \overrightarrow{X} = \overrightarrow{H} \% x^m = \frac{\overrightarrow{H}(1 - \overrightarrow{a}) \% x^m}{1 - \overrightarrow{a}} = \frac{(\overrightarrow{H} \% x^m (1 - \overrightarrow{a}) \% x^m) \% x^m}{(1 - \overrightarrow{a}) \% x^m}$$

$$\text{注：上式分子实际上是个有穷多项式，特别当心} \overrightarrow{a} \text{下标是从} 1 \text{开始的，即} [1, m]$$

当然，上述方法适用于知道x_0，比较快速地求出$f[1..n], k \leq n \leq 10^6$

更本质地讲，如果我们把所有序列下标为负的部分，都看作0，那么其实x_0不再是序列x的首项

这样讲的意义在于，我们可以重构序列$\overrightarrow{X'} = [x_i], i \in [-(m-1), 0]$

从而重构：$\overrightarrow{x'}^T = [x_0, x_{-1}, x_{-2}, \ldots, x_{-(m-2)}, x_{-(m-1)}]$

我们仍然左乘A^n取结果向量第1个元素，亦即：

$$x_n = (A^n \overrightarrow{x'})[1] = \sum_{i=0}^{m-1} c_i (A^i \overrightarrow{x'})[1] = \sum_{i=0}^{m-1} c_i ((A^i \overrightarrow{x'})[1]) = \sum_{i=0}^{m-1} c_i x_i = \overrightarrow{C} \cdot \overrightarrow{X}$$

$$\text{其中：} \overrightarrow{C} = [c_i], \overrightarrow{X} = [x_i], i \in [0, m-1], \text{另 } \cdot \text{表示向量点积}$$

发现最终需要的还是$\overrightarrow{X}$，而不是$\overrightarrow{X'}$；我们左乘A^{n+m-1}取结果向量第m个元素，亦即：

$$x_n = (A^{n+m-1} \overrightarrow{x'})[m] = \sum_{i=0}^{m-1} c'_i (A^i \overrightarrow{x'})[m] = \sum_{i=0}^{m-1} c'_i ((A^i \overrightarrow{x'})[m]) = \sum_{i=0}^{m-1} c'_i x_{i-m+1} = \overrightarrow{C'} \cdot \overrightarrow{X'} = c'_{m-1} x_0$$

$$\text{其中：} \overrightarrow{C'} = [c'_i], i \in [0, m-1], \overrightarrow{X'} = [x_i], i \in [-(m-1), 0], \text{另 } \cdot \text{表示向量点积}$$

因此可以看到，我们不再需要计算多余的前m项内容，只要计算多项式快速幂模，亦即$\overrightarrow{C'}$即可

推广：

求递推序列$\{x_i\}$的前$n+1$项和s_n，沿用上述的方法：

令C_i表示多项式x^i对特征多项式G取模，令$C = \sum_{i=0}^{n} C_i = (\sum_{i=0}^{n} x^i) \% G$，类似于快速幂的递归算法可以求解$C$，结果仍然是$\overrightarrow{C} \cdot \overrightarrow{X}$

【93】概率论：全期望公式

$$E(E(X|Y)) = E(X)$$

$$\text{注意：} E(X|Y) \text{是个随机变量，概率分布：以} P(Y = y) \text{概率取到} E(X|Y = y)$$

进一步分析可见维基百科或者2013年国家集训队论文《概率论应用》

【94】组合：超平面切平面的规律

经典问题：n个$k-1$维超平面最多可以将k维空间分割成多少个空间区域

$$f(n,k)=\sum_{i=0}^{min(k,n)}\binom{n}{i}$$

$f(n,k)$	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	...
0	1	1	1	1	1	1	1	1	1	...
1	1	2	2	2	2	2	2	2	2	...
2	1	3	4	4	4	4	4	4	4	...
3	1	4	7	8	8	8	8	8	8	...
4	1	5	11	15	16	16	16	16	16	...

递推公式：

$$f(0,k)=1,k\geq 0$$
$$f(n,k)=f(n-1,k-1)+f(n-1,k)$$

显然将上表中，每一行都差分一下，记得到组合数表；可以这样理解，每个第0行的1都为下面区域带来一个组合数的贡献，那么(n,k)相当于得到了行总是n但是列$\in[0,k]$的组合数贡献之和

该序列可以通过莫队算法或者分治fft快速求得一组，线性时间可预处理一行或者一列

特别的，令$k=2$有：直线分割平面的区域数目：

$$f(n,2)=\binom{n}{0}+\binom{n}{1}+\binom{n}{0}=n(n+1)/2+1$$

一些其他问题，比如折线划分平面，或者闭曲线椭圆划分平面，这些都可以用类似的类比递推归纳的方法，得到递推式，但暂时不便于推广到高维情形

【95】代数/拓扑：全序关系、偏序关系与偏序集

全序关系$\leq$满足：完全性，即$\forall x,y\in S,x\leq y||y\leq x$

偏序关系$\leq$满足：自反性(反身性)，即$\forall x\in S,x\leq x$

两者均满足：

反对称性，即$\forall x,y\in S,x\leq y\&\&y\leq x\leftrightarrow x=y$

传递性，即$\forall x,y,z\in S,x\leq y\&\&y\leq z\rightarrow x\leq z$

由于完全性是反身性的特例，故全序关系是偏序关系的特例，即全序关系一定是偏序关系

二者直观的区别在于，偏序关系并不是两两可比(默认只知道自己和自己可比)，但是全序关系是两两可比的

全序集及其笛卡尔积偏序线性空间中的min/max函数的基本性质与推广

- 都满足交换律，结合律等基本性质
- 满足对称分配率，即min对max满足分配率，反之亦然

设全序集$(S,\leq)$中运算$+$表示二元min函数，运算$*$表示二元max函数，该表示可以用多元多项式刻画复杂的min/max嵌套表达式

令$T_n=S^n,n\in[1,+\infty]$，注意$(T_n,\leq)$一般不再是全序集，但是如果定义T_n上的min为各维min的笛卡尔积，可以发现仍然满足上述的分配率；当$S=\{0,1\}$时，$\&$为S^n上的min，$|$为S^n上的max；当$S=N$时，gcd为S^n上的min，lcm为其max，可以推广到$S=Z$，涉及到有理数的gcd；当S^k为全集$U(|U|=n)$的k元子集簇时，$\cap$为其min，$\cup$为其max

与gcd有关的分配率的直接推论：

- $lcm(gcd(x,y),gcd(x,z),gcd(y,z))=gcd(lcm(x,y),lcm(x,z),lcm(y,z))$
- n元整数集$(k-k)gcd$的lcm等于$((n-k+1)-(n-k+1))lcm$的gcd

【100】数论/密码学：RSA公钥体系

公钥加密，私钥解密，RSA可靠性取决于大数因式分解的困难性

取两个大而不太接近的质数p,q，令$n=pq,t=\phi(n)=(p-1)(q-1)$

取一个随机数$e\in[2,t),gcd(e,t)=1$，令$d=e^{-1}\pmod t$

(n,e)为公钥，(p,q,d)为私钥，明文为M，密文为C

加密算法：$C=M^e\pmod n$

解密算法：$M=C^d\pmod n$

【101】数论/密码学：$ElGamal$密码

其可靠性取决于大数离散对数的困难程度

选择一个大质数p，且要求$p-1$含大质因子，选择一个p的生成元或原根a

随机选择一个$d\in[2,p-2]$，令$y=a^d\pmod p$

公开p,a，取y为公钥，d为私钥，明文为M，密文为(C_1,C_2)

加密算法：随机选取$k\in[2,p-2]$

$C_1=a^k\pmod p$

$C_2=My^k\pmod p$

解密算法：$M=C_2C_1^{-d}\pmod p$

【106】数论：$Vantieghemstheorem$

n是质数当且仅当：

$$\prod_{1\le k\le n-1}\left(2^k-1\right)\equiv n\pmod{2^n-1}$$

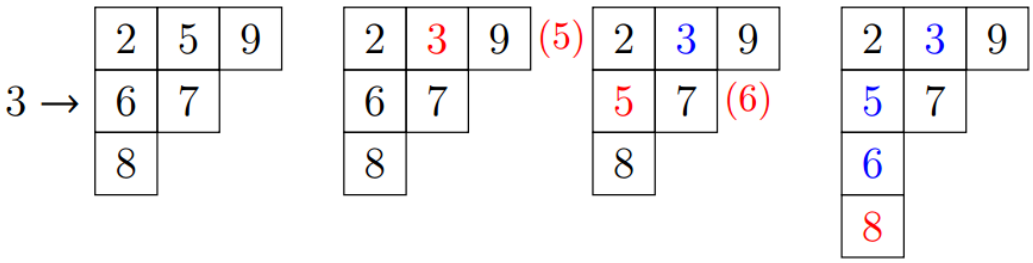
可以换成其他的公式：

$$\prod_{1\le k\le n-1}\left(X^k-1\right)\equiv n-\left(X^n-1\right)/(X-1)\pmod{X^n-1}$$

$$\prod_{1\le k\le n-1}\left(X^k-1\right)\equiv n\pmod{\frac{X^n-1}{X-1}}$$

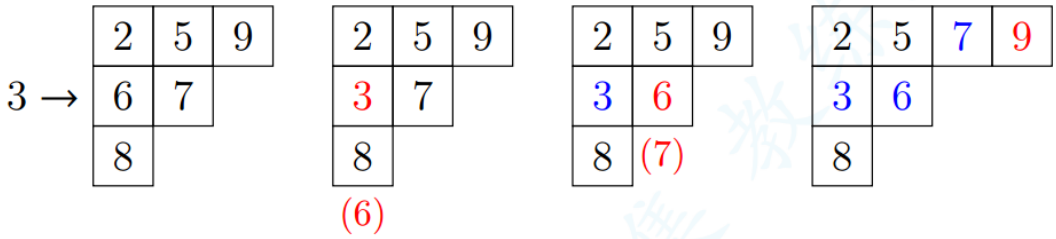
【118】组合/代数：杨氏矩阵/杨表(Young tableaux) 维基百科：杨表

- 定义
  - **形状**：N的一个不严格降序拆分称为形状，$\lambda=(\lambda_1,\lambda_2,\lambda_3,\ldots,\lambda_m)$，$\lambda_i\ge\lambda_{i+1}$，$\lambda\vdash N$
  - **杨图**：某种形状λ的未填数字的表格，共m行，第i行λ_i列
  - **杨表**：填满数字的杨图
    - **标准杨表**：从上向下严格增，从左往右严格增，$1..N$各出现一次
    - **半标准杨表**：从上向下严格增，从左往右不严格增
    - **杨表权重**：一个向量$(1,1,\ldots,1)$表示各数字出现次数
  - **斜杨图/表**：λ/μ，$\lambda_i\ge\mu_i$，第i行在$[\mu_i+1,\lambda_i]$列填数字
    - 标准/半标准：参见上面
  - **边角**：一个格子为边角，其下面和右边的格子不存在
- 杨表与排列
  - 行插入算法$S\leftarrow x$
    - 所移动的格子看上去一定是向不严格的左下方移动

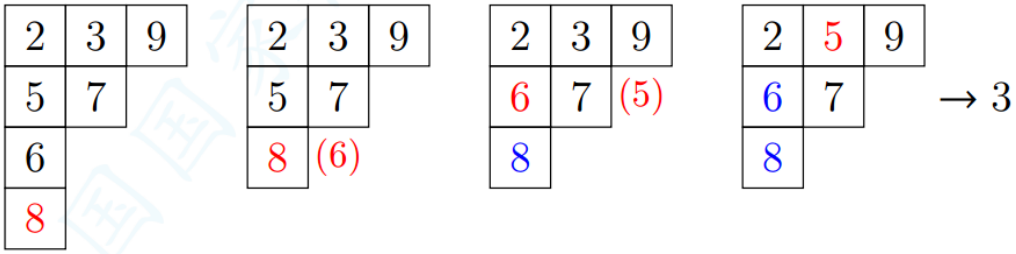


  - 列插入算法$x\rightarrow S$

- 与行插入对称



- 边角删除算法
  - 与行插入互为逆操作



- 记录表：一个排列X进行行插入得到杨表P_X，同时可以得到记录表Q_X，只需要每次在将元素x_i插入时，杨表增加新元素的位置，放入下标i；记录表Q_X与P_X同规模

$$X=1,5,3,2,6,7,4\rightarrow \qquad P_X=\begin{array}{|c|c|c|c|}\hline 1&2&4&7\\\hline 3&6\\\cline{1-2}5\\\cline{1-1}\end{array} \qquad Q_X=\begin{array}{|c|c|c|c|}\hline 1&2&5&6\\\hline 3&7\\\cline{1-2}4\\\cline{1-1}\end{array}$$

- 一一对应：Robinson–Schensted correspondence
  - 长度为n的排列与**一对形状相同的标准杨表**一一对应
  - 令f_λ表示形状为λ的标准杨表的数量，则$\sum_{\lambda\vdash n} f_\lambda^2=n!$
- 排列插入算法的性质
  - 行列交换律：$(x\rightarrow S)\leftarrow y=x\rightarrow(S\leftarrow y)$，这表明相邻两个对行列的插入操作(一个对行，一个对列)是可交换的
  - 行列相对独立性：一个插入操作序列，可以调整为先完成行操作，再做列操作，每种相对顺序不变
  - 翻转特性：X^R表示排列的左右翻转，则$P_X=(P_{X^R})^T$，这表明排列行插入等价于倒序列插入，注意只对杨表成立，对记录表**不**成立
  - 相反数特性：排列倒序行插入得到的杨图(形状)等价于原排列***相反数序列***行插入得到的杨图，注意不是杨表相同
- 杨表与对称矩阵
  - 普通非负整数矩阵
    - 矩阵A拆成序列对X：其中(i,j)这个数对出现$A[i][j]$次，数对按照$pair$排序，即构成$(first,second)$序列对
    - X^{-1}：只要把每个数对两维交换，重新排序，得到新的序列对
    - P_X表示将X第二行(即第二维，或者$second$序列)插入构造的杨表，Q_X表示在P_X构造过程中，将$first$序列视为下标构造的记录表
    - 性质关系：$(P_X,Q_X)=(Q_{X^{-1}},P_{X^{-1}})$

$$X=\begin{pmatrix}2&3&4&5&5&5&6\\1&5&3&2&6&7&4\end{pmatrix}\rightarrow \qquad P_X=\begin{array}{|c|c|c|c|}\hline 1&2&4&7\\\hline 3&6\\\cline{1-2}5\\\cline{1-1}\end{array} \qquad Q_X=\begin{array}{|c|c|c|c|}\hline 2&3&5&5\\\hline 4&6\\\cline{1-2}5\\\cline{1-1}\end{array}$$

$$X^{-1}=\begin{pmatrix}1&2&3&4&5&6&7\\2&5&4&6&3&5&5\end{pmatrix}\rightarrow \qquad P_{X^{-1}}=\begin{array}{|c|c|c|c|}\hline 2&3&5&5\\\hline 4&6\\\cline{1-2}5\\\cline{1-1}\end{array} \qquad Q_{X^{-1}}=\begin{array}{|c|c|c|c|}\hline 1&2&4&7\\\hline 3&6\\\cline{1-2}5\\\cline{1-1}\end{array}$$

  - 对称矩阵
    - 一些定义：矩阵$M_{W\times H}$，行和r_i，列和c_j，X_M为其所构序列对，$P_M=P_{X_M}$，同理定义Q_M，$X_M^{-1}=X_{M^T}$
    - 一一对应：M和一对半标准杨表P_M和Q_M一一对应，P_M中i出现c_i次，Q_M中i出现r_i次；特别的，M为对称矩阵，与半标准杨表P_M一一对应，P_M中i出现r_i次，且$X_M^{-1}=X_M$，$P_M=Q_M$

        - 对称矩阵M的迹$tr(M)=\sum_{i=1}^{n}M_{ii}$等于$P_M$中长度为奇数的列数
    - 对合排列(*involution permutation*)
        - 定义：对合排列$X=X^{-1}$，即X的圈结构中，圈长非1即2；对应着每行每列只有一个1的01矩阵
        - 性质：其自环个数等于其所对应的标准杨表中奇数长度的列的个数
        - 对合数：$a_0=a_1=1,a_n=(n-1)a_{n-2}+a_{n-1}$
- 杨表与最长上升/下降子序列
    - 基本结论：X为一个排列
        - P_X第一行长度为$len(LIS_X)$，第一列长度为$len(LDS_X)$
    - 最长$k-LIS$子序列
        - 定义：长度不超过k的LIS子序列为$k-LIS$子序列
        - 结论：排列X的最长$k-LIS$子序列长度等于杨表P_X的前k列长度之和
        - 例题：$CTSC2017$最长上升子序列，询问Q次排列前缀的最长$k-LIS$子序列长度(k变化)；只维护杨表前$\sqrt{n}$行和$\sqrt{n}$列，前$\sqrt{n}$行，通过插入相反数序列维护杨图；复杂度$O(n\sqrt{n}logn)$
    - LIS计数
        - 结论
            - 长度为n的LIS长度为α，且LDS长度为β的排列个数为：
            $$num(n,\alpha,\beta)=\sum_{\text{行数为}\beta,\text{列数为}\alpha,\lambda\vdash n}f_\lambda^2$$
            - 权值范围为$[1,n]$，各权值数量分布为μ，即：$\sum_{i=1}^{n}\mu[i]=n$；令$g_{\lambda/\mu}$表示形状为λ且权值分布为μ的**半标准**杨表数量；则这样的长度为n的序列中，最长**非严格**上升子序列长度为α且最长**严格**下降子序列长度为β的序列数量为：
            $$num(n,\alpha,\beta)=\sum_{\text{行数为}\beta,\text{列数为}\alpha,\lambda\vdash n,\mu}g_{\lambda/\mu}f_\lambda$$
        - 例题：$BJWC2018$最长上升子序列
            - 题目：长度为n的随机排列X，求$E(len(LIS_X))$
            - 分析：枚举n的所有降序拆分λ，用钩子公式$O(n^2)$计算f_λ，复杂度$O(n^2p(n))$，可轻松做到$n\le 63$
- 杨表与钩子公式
    - 钩子函数：$h_\lambda(i,j)$表示在形状为λ的杨图中(i,j)右侧和下侧以及自己所确定的一块钩子区域的格子数量
    - 标准杨表钩子公式：
    $$f_\lambda=\frac{n!}{\prod h_\lambda(i,j)}$$
    $$f_\lambda=n!\frac{\prod_{1\le j<k\le m}((\lambda_j-j)-(\lambda_k-k))}{\prod_{i=1}^{m}(\lambda_i+m-i)!}$$
    注：上面的公式更便于理解和记忆，下面的公式更便于计算和实现，时间复杂度为$O(m^2)$
- 杨图随机游走
    - 定义：从杨图上某个格子出发，随机向右边或下边走一格，直到走到边角停止
    - 等概率随机游走，$\lambda=(\lambda_1,\lambda_2,\cdots,\lambda_m)$
    $$p(\ (1,1)\to\text{边角}(r,s)\ )=\frac{1}{n}\prod_{i=1}^{r-1}\frac{h_\lambda(i,s)}{h_\lambda(i,s)-1}\prod_{j=1}^{s-1}\frac{h_\lambda(r,j)}{h_\lambda(r,j)-1}$$
    - 带权随机游走，$\lambda'_i=|\{j|\lambda_j\ge i\}|$，$x_i,y_i$为每列/每行的权重
    $$p(\ (1,1)\to\text{边角}(r,s)\ )=$$
    $$\frac{x_ry_s}{\sum_{(p,q)\epsilon[\lambda]}}\prod_{i=1}^{r-1}\left(1+\frac{x_i}{x_{i+1}+\cdots+x_r+y_{s+1}+\cdots+y_{\lambda_i}}\right)$$
    $$\times\prod_{j=1}^{s-1}\left(1+\frac{y_j}{x_{r+1}+\cdots+x_{\lambda'_j}+y_{j+1}+\cdots+y_s}\right)$$
- 杨表与网格图路径
    - 卡特兰数相关
        - $2\times n$的标准杨表个数为卡特兰数：$C_n=\binom{2n}{n}/(n+1)$
        - 分析方法：对每一维坐标，记录为杨表的一行，x_{ij}表示i这一维坐标，第一次到达j的时刻；考虑一条合法的路径，每个时刻会且仅会使得一维坐标达到新的值，当走到终点$(a_1,a_2,\cdots,a_m)$时，每维均取到了终点坐标的上界，因此我们得到的是一个m行，$\lambda=\overrightarrow{a}$的标准杨表；考虑杨表的限制，每列递增，表示不同维到达同一个值的时刻偏序关系，对应到路径中任何时刻走到的点的坐标应该满足逐维递减；例如卡特兰数要求路径每点横坐标不小于纵坐标，即不穿过对角线
        - 例题：$LOJ6051$PATH
            - 题目：m维空间，从原点出发，到达$\overrightarrow{a}$，每次某一维坐标增加1，问随机路径中能够保持坐标逐维不升的概率，保证$\overrightarrow{a}$满足此性质；即求***高维卡特兰数***

- 分析：令$r_i=\lambda_i+m-i$，$n=\sum_{i=1}^m a_i$，由钩子公式：

$$ans=\frac{\frac{n!}{\prod_{i=1}^m r_i!}\prod_{1\le j<k\le m}(r_j-r_k)}{\frac{n!}{\prod_{i=1}^m a_i!}}=\left(\prod_{i=1}^m\frac{a_i!}{r_i!}\right)\prod_{1\le j<k\le m}(r_j-r_k)$$

  注：最右边的公式可以用ntt优化到$mlogm$，这是一个差卷积

- 不交叉网格路径
  - Lindström–Gessel–Viennot lemma($LGVL$)

    该定理是说：边带权DAG中有两个点集$A,B,|A|=|B|=n$，我们枚举$A\to B$的$n!$个对应关系，对每种对应关系(一个排列σ)，对每对对应点，枚举一条有向路径，且这n条路径不交，令$\omega(P)$表示路径边权之积，则累乘n条不交路径的权，最后叠加到答案中(正负取决于σ逆序对)，该答案等于下面矩阵的行列式的值

$$M=\begin{pmatrix}e(a_1,b_1)&e(a_1,b_2)&\cdots&e(a_1,b_n)\\e(a_2,b_1)&e(a_2,b_2)&\cdots&e(a_2,b_n)\\\vdots&\vdots&\ddots&\vdots\\e(a_n,b_1)&e(a_n,b_2)&\cdots&e(a_n,b_n)\end{pmatrix}$$

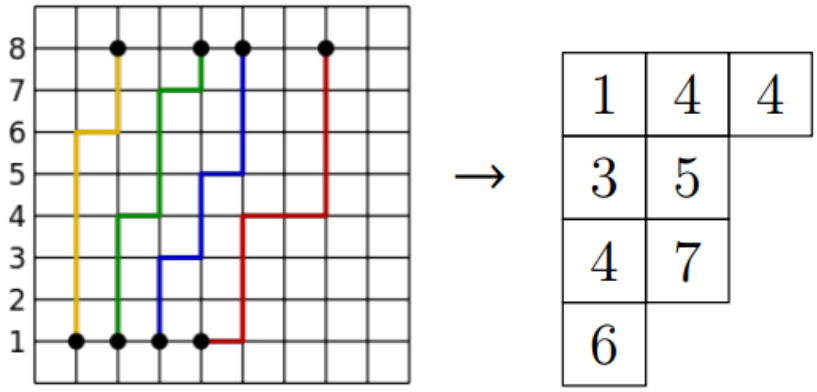
$$\det(M)=\sum_{(P_1,\ldots,P_n):A\to B}\operatorname{sign}(\sigma(P))\prod_{i=1}^n\omega(P_i)$$

    其中$e(a,b)=\sum_{P:a\to b}\omega(P)$，即所有$a\to b$的有向路径权值积的和

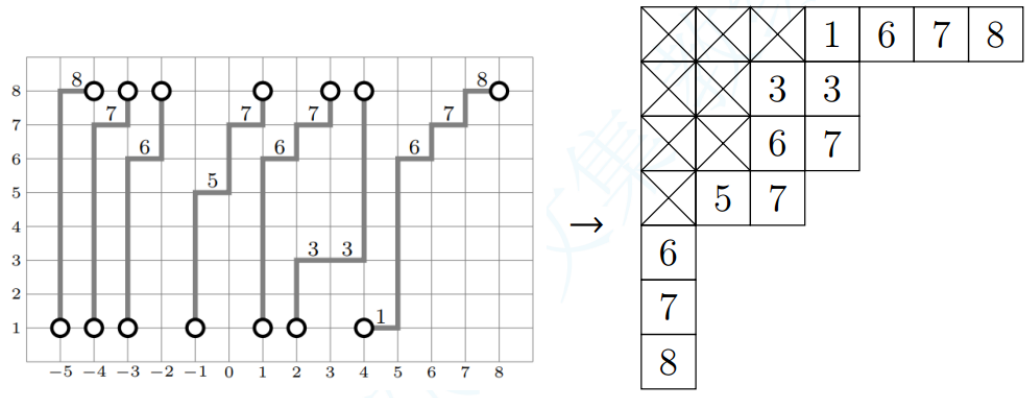
    特别的，若图的边权均为1，则该行列式的值即为$A\to B$的不交路径元祖计数

    注意上述的乘法，可以换成一些可重载的运算，如字符串的连接操作等

    参考阅读：1 文献
  - $LGVL$用于网格图
    - 如果是网格图，只允许向右向上走，不过是特殊化的DAG，带入$E(x,y)$即可
    - 由于在此意义下，所枚举到的排列σ，只有一种是合法的，因此适当排序之后，上述行列式值直接等于$A_i\to B_i$所构成的n元组路径方案；形式化的，我们要求A,B中点横坐标不升，纵坐标不降
  - 一一对应
    - 权值$\in[1,n]$且m行($n=\sum_{i=1}^m\lambda_i$)的半标准杨表与$A=\{(-i,1)\},B=\{(\lambda_i-i,n)\},i\in[1,m]$的$m$条不交路径元组一一对应；构造方法：每条路径对应杨表一行，该路径每右行一步，记录其行号



    - 权值$\in[1,n]$且m行($n=\sum_{i=1}^m\lambda_i$)的半标准斜杨表λ/μ与$A=\{(\mu_i-i,1)\},B=\{(\lambda_i-i,n)\},i\in[1,m]$的$m$条不交路径元组一一对应(图表右移了两格)



- 行列式公式
  - 标准斜杨表数量公式

$$f_{\lambda/\mu}=\left(\sum_i \lambda_i-\mu_i\right)!\left|\frac{1}{(\lambda_i-i-\mu_j+j)!}\right|_{i,j=1}^{|\lambda|}$$

- 注：负数的阶乘倒数定义为0

- 标准杨表数量公式

$$f_\lambda=\left(\sum_i \lambda_i\right)!\left|\frac{1}{(\lambda_i+j-i)!}\right|_{i,j=1}^{|\lambda|}$$

- 欧拉数：长度为$2n$的波形排列计数

$$(2n)!\det\left(\frac{1}{(2j-2i+2)!}\right)_{i,j=1}^{n}$$

- 思路：构造一个斜杨表$(2n-i+2)/(2n-i), i\in[1,m]$，填好数之后从下而上，从左向右取数构造排列

- Hankel 矩阵

- 定义：$Hankel$矩阵每条副对角线值相等，由序列a生成的矩阵如下：

$$A=\begin{bmatrix} a_0 & a_1 & a_2 & \cdots & \cdots & a_{n-1} \\ a_1 & a_2 & & & & \vdots \\ a_2 & & & & & \vdots \\ \vdots & & & & & a_{2n-4} \\ \vdots & & & & a_{2n-4} & a_{2n-3} \\ a_{n-1} & \cdots & \cdots & a_{2n-4} & a_{2n-3} & a_{2n-2} \end{bmatrix}$$

- $Hankel$变换：由b序列变换为h序列

$$h_n=\det\left(b_{i+j-2}\right)_{1\le i,j\le n+1}$$

$$令\ c_n=\sum_{k=0}^{n}\binom{n}{k}b_k$$

$$则有：\ \det\left(b_{i+j-2}\right)_{1\le i,j\le n+1}=\det\left(c_{i+j-2}\right)_{1\le i,j\le n+1}$$

$$即\ h在原序列二项式变换下，保持不变$$

- 相关定理

- $$\begin{aligned}&\det\left(\left(X_i+A_{n-1}\right)\cdots\left(X_i+A_{j+1}\right)\left(X_i+B_j\right)\cdots\left(X_i+B_1\right)\right)_{i,j=0}^{n-1}\\&=\prod_{0\le i<j\le n-1}\left(X_i-X_j\right)\prod_{1\le i\le j\le n-1}\left(B_i-A_j\right)\end{aligned}$$

- $$\det\left(C_{\alpha_i+j}\right)_{i,j=0}^{n-1}=\prod_{0\le i<j\le n-1}\left(\alpha_j-\alpha_i\right)\prod_{i=0}^{n-1}\frac{(i+n)!\,(2\alpha_i)!}{(2i)!\alpha_i!\,(\alpha_i+n)!}$$

- $k-Dyck\ Path$

- $Dyck\ Path$是说一条从$(0,0)$到$(2n,0)$的一条路径，每次横坐标增加1，纵坐标增加1或-1，且不越过x轴

- $k-Dyck\ Path$是说有k条这样的路径，第一条不越过x轴，第i条不越过即不低于第$i-1$条路径

- 一一对应：列数不超过$2k$的，元素$\in[1,n]$内的且**每行长度为偶数**的**半标准杨表**和长度均为$2n+2$的$k-Dyck\ Path$构成一一对应；若将其旋转一下，可转化为从$(0,0)$到$(n+1,n+1)$的k条路径，每次只可以向右向上走一步，不能越过上一条路径，第一条不低于$y=x$

- 计数公式

$$b_{n,k}=\prod_{1\le i\le j\le n}\frac{2k+i+j}{i+j}$$

- 下三角矩阵的LDS

- 下三角矩阵M的LDS定义为权值和最大的路径，可选一个起点，然后每次走向右上方的某个格子，或者同行的右侧的某个格子，注意不要求权值递增

- $LDS(M)\le k$的下三角矩阵M与$k-Dyck\ Path$一一对应

- 元素$\in[1,n]$，每行长度为偶数，半标准杨表和$n\times n$的对称矩阵M'一一对应，杨表列数$=LDS(M')$

- 元素$\in[1,n]$，列数$\le k$半标准杨表与$n\times n$元素$\in[0,k]$满足每行每列均非严格递增的**对称矩阵**数量相等

$$a_{n,k}=\prod_{1\le i\le j\le n}\frac{k+i+j-1}{i+j-1}$$

- 半标准杨表计数

- 元素$\in[1,n]$，半标准杨表$\lambda=(\lambda_1,\cdots,\lambda_m), n=\sum_{i=1}^{m}\lambda_i$的计数公式

$$\prod_{(i,j)\in\lambda}\frac{n+j-i}{h_\lambda(i,j)}=\prod_{1\le i<j\le n}\frac{\lambda_i-\lambda_j+j-i}{j-i}$$

- 例题：$CodeChef\ BillBoards(BB)$

- 形状带限制的杨表计数
  - 列数$\leq k$，即$LIS \leq k$的排列计数
    - 定义

$$u_k(n) = \sum_{\lambda \vdash n, \lambda_1 \leq k} f_\lambda^2$$

$$U_k(x) = \sum_{n \geq 0} u_k(n) \frac{x^{2n}}{n!^2}$$

$$I_i(2x) = \sum_{n \geq 0} \frac{x^{2n+i}}{n!(n+i)!}$$

其中$I_k(2x) = I_{-k}(2x)$表示第一类修正贝塞尔函数，有许多性质

    - 定理

$$U_k(x) = \det\left(I_{i-j}(2x)\right)_{i,j=1}^K$$

$$U_2(x) = \begin{vmatrix} I_0(2x) & I_1(2x) \\ I_1(2x) & I_0(2x) \end{vmatrix} = I_0(2x)^2 - I_1(2x)^2$$

$$\Rightarrow u_2(n) = \frac{1}{n+1}\binom{2n}{n}$$

$$u_3(n) = \frac{1}{(n+1)^2(n+2)} \sum_{j=0}^{n} \binom{2j}{j} \binom{n+1}{j+1} \binom{n+2}{j+2}$$

  - 行数$\leq k$
    - 定义

$$y_k(n) = \sum_{\lambda \vdash n, \lambda_1 \leq k} f_\lambda$$

    - 定理

$$y_2(n) = \binom{n}{\lfloor n/2 \rfloor}$$

$$y_3(n) = \sum_{i=0}^{\lfloor n/2 \rfloor} \binom{n}{2i} C_i$$

$$y_4(n) = C_{\lfloor (n+1)/2 \rfloor} C_{\lceil (n+1)/2 \rceil}$$

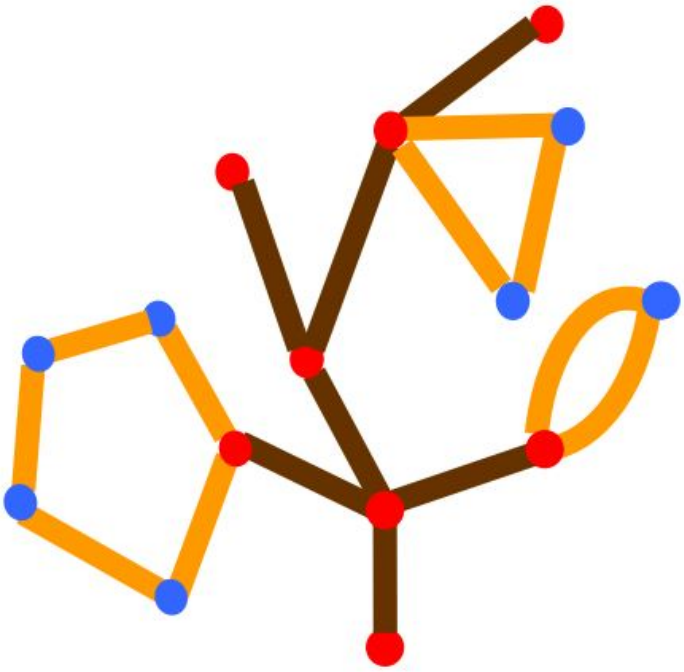
$$y_5(n) = 6 \sum_{i=0}^{\lfloor n/2 \rfloor} \binom{n}{2i} C_i \frac{(2i+2)!}{(i+2)!(i+3)!}$$

【122】博弈论：基础知识梳理

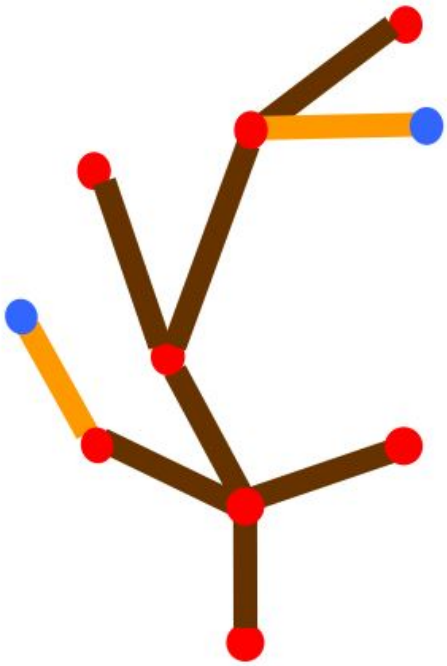
- NP状态定理
- sg函数理论
  - 适用组合游戏特点
    - 二人博弈，双方绝对聪明且利己
    - 游戏在有限步内终止，无法决策方输
    - 游戏状态不能重复抵达，游戏无平局；即游戏图为DAG
    - 决策空间只与游戏状态有关，与游戏者无关，即游戏决策是对称的，游戏是信息公开的
  - sg函数定义与递推
  - 游戏和与nim和
- 普通Nim游戏
  - 一步策略：若n堆依次为$x_1, x_2, \ldots, x_n$，异或和不为0，先手存在必胜策略；下面构造一个方案，令$X = XOR_{i=1}^n x_i$，只需要将某个x_i变成$x_i \oplus X < x_i$即可；考虑寻找一个$i \in [1, n]$，满足x_i在与X对其的最高位为1，即满足前述要求
- $Anti-Nim$游戏与SJ定理
  - 定义：nim游戏中，拿走最后一个石子者输，即决策空间为空者赢
  - 结论：先手必胜当且仅当：
    - 游戏的$sg > 0$，且存在单一游戏$sg > 1$
    - 游戏的$sg = 0$，且所有单一游戏$sg \leq 1$
- $Multi-Nim$游戏
  - 定义：可以将一堆石子分成若干堆(可指定或者不指定堆数)，即一个单一游戏的后继可以为多个单一游戏的和
  - 结论：利用经典的sg递推即可，不同决策的后继之间用mex操作，游戏和用nim和
- $Every-Nim$游戏

- 定义：每一轮游戏，选手必须将每个没有结束的单一游戏，推进一步；必胜则要求最长的步数使得到达胜利状态，必败则要求最短的步数使得到达必败态
- 令$step[x]$表示状态x的步数，满足上述所求
  - 若v为终止状态，$step[v]=0$
  - 若$sg[v]>0$，$step[v]=max\{step[u]\}+1$，$sg[u]=0$，$v\to u$
  - 若$sg[v]=0$，$step[v]=min\{step[u]\}+1$，$v\to u$
- 结论：先手必胜当且仅当单一游戏的最大$step$为奇数

- 阶梯博弈
  - 定义：有一组阶梯编号为$0,1,2,\dots$；每个阶梯上有一些石子，每次可以选择一个阶梯移动一部分到左边紧靠着的阶梯，不能移动者输；0为地面。
  - 结论：等价于只考虑奇数编号的阶梯的nim游戏，因为对方将奇数移动到偶数，看成某一堆石子减少，将偶数移动到奇数，则你可以立刻将同样多的石子从奇数移动到偶数，故操作抵消，奇数编号石子数不变。
- $nimk$游戏
  - 定义：n堆石子，双方轮流取，每次可以取最多k堆，每堆可以取任意个石子(至少1个)，无子取者败
  - 结论：将每堆石子数a_i转化为二进制数，统计每一位上的1个数之和，若每一位的和都是$k+1$的倍数，则先手必败，否则先手必胜
- $Bash$博弈及其扩展
  - 普通$Bash$博弈
    - 定义：每次选择一堆，取走数量$\in[1,m]$
    - 结论：n为$m+1$倍数为必败态，否则为必胜态
  - $Bash$博弈扩展
    - 定义：每次选择一堆，取走数量$\in[a,b]$
    - 结论：$n\in[1,a)\cup(b,a+b)$平局，除此之外$n\%(a+b)=0$先手必败，否则先手必胜
- 硬币游戏
  - 定义：按照规则翻一系列硬币，要求满足所翻动的下标最大的硬币必须从正面到反面
  - 基本结论：局面的sg值等于局面中正面朝上即1单独存在时的单一游戏sg值的nim和
  - 已知经典游戏：下标从1开始
    - 每次翻连续k个
      - sg值分布为：$00...001$周期为k，即$sg(x)=[k|x]$
    - 每次先翻动x，然后在$x-1,x-2,x-3$中选择一个翻动
      - sg值分布为：$1,2,3,0,1,2,3,0...$，周期是4
    - 每次翻动严格2个，下标差$\in\{1,2,3\}$，下标从0开始
      - sg值分布为：$0,1,2,3,0,1,2,3,\dots$，注意$sg(0)=0$，周期为4
    - 每次翻动个数不超过3个，下标从0开始
      - sg值分布为：$sg(x)=2x+!parity(x)$，$parity(x)$表示x二进制1的个数是否为奇数
    - 每次翻动连续任意多个，至少1个，下标从1开始(尺子游戏)
      - sg值分布为：$sg(x)=lowbit(x)=x\&-x$
- 删边游戏
  - 树上删边游戏
    - 定义：给定n个点的有根树，双方轮流删边，移走不与根连通的分支，无法操作者输
    - 结论：叶子节点sg值为0，非叶节点的sg值等于儿子的$sg+1$的异或和
    - 分析：某棵子树x可以等效的替换为长度(边数)为$sg(x)$的单一链分支，那么考虑很多儿子的子树，他们需要先生长一条边，然后并联时，直接作为游戏和异或合并；等价于转化为一堆石子数为x的单一游戏，$sg(x)=x$
  - 特殊图上删边游戏：$Christmas\ Game$
    - 定义：图可看成从基础树，添加一些环得到；每次选择树上一个点，在空中生长出一个环还要回到这个点，不能经过其他树上的点



- 结论：偶环直接去掉，奇环等效为长度为1的链



- 分析：长度为偶数的环，无论剪去哪条边，都只能分成一奇一偶两条链，其异或和为奇数，故$sg=0$，可消去；或者说，无论先手删哪一条边，对方可以对称的删对应边，先手必败，$sg=0$；长度为奇数的环，首先先手是必胜的，因为可以删正中间的边，然后对称操作即可，故$sg>0$，而删去一条边后得到奇偶性相同的两条链，其异或和为偶数，故$sg=1$

- 图上删边游戏
  - 定义：无向连通图，有一个点为根，不能删边者输，其余规则一样
  - 结论：$Fusion\ Principle$
    - 偶环缩成新点，所有与环相连的边直接与新点相连；奇环缩成新点，延伸一条新边；这样转换sg值不变

- 有环sg博弈/有向图博弈
  - 使用$spfa$更新NP状态值或者sg函数值
- 三人nim游戏
- $K-L-nim$游戏
- k倍动态减法游戏
  - 定义：有一堆石子，含m个，双方轮流取，不能取输；第一个人可以取任意个，不能一次取完；以后每次取的石子数量不超过上一次的k倍(严格小于m个)
  - 结论：

- 特殊情况：$k=1$，当且仅当m为2的幂次时，后者胜利，否则先手胜利；$k=2$游戏退化为斐波那契博弈
- 一般情况：令$NP(x,y)$表示当前有x个石子，本次所取不超过y个，是否先手必胜；显然存在单调性，即若$NP(x,y_0)=1$，则$\forall y\geq y_0, NP(x,y)=1$；故令$f(x)$表示石子个数为$x$时，使得$NP(x,y_0)=1$的最小$y_0$；考虑画出$NP$表格图，值为0的一行中连成一堵墙，如图所示：

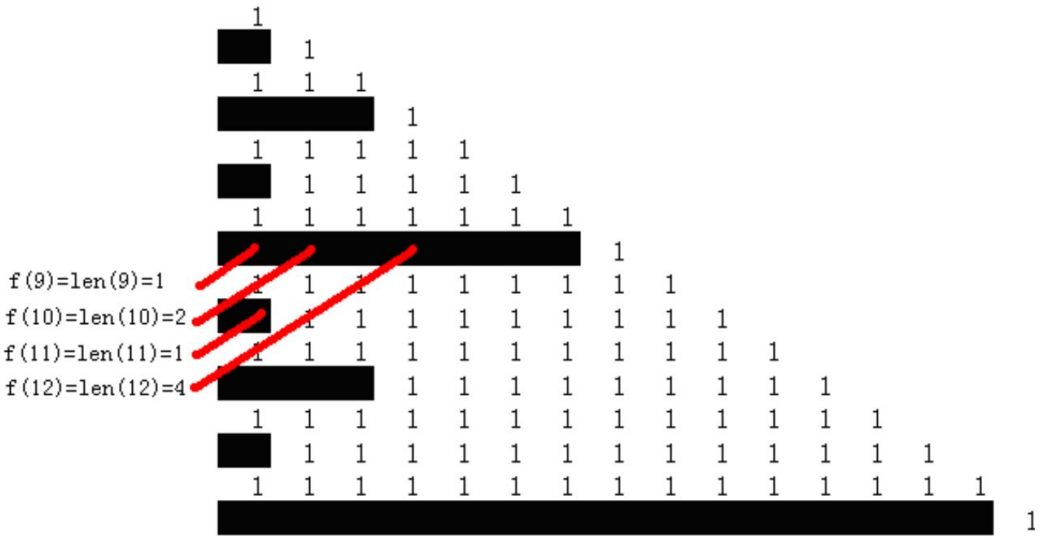


图4-2　k=1时NP(m,n)的情况　　NP(m,n)=0的连续的格子就像一堵堵墙

考虑一行一行求解$f(i)$，i为行数；$f(i)$具有几何意义：从$(i,0)$这个格子，沿着直线$y=x$出发，撞到第一赌墙，所走的步数；注意墙有单调性，即若一堵墙不能挡住当前的射线，则再也不可能挡住以后产生的射线；考虑采用单调栈，从栈顶开始，挡不住的墙依次弹栈，直到撞到，计算步数即$f(i)$，然后产生长度为$f(i)-1$的新墙，压栈；时间复杂度$O(m)$，从几何角度不难实现；先手必败当且仅当$f(m)=m$，因为先手第一次只能取$[1,m-1]$内的石子个数，即$NP(m,m-1)$

- 斐波那契博弈
  - 定义：一堆石子，先手不能全取完，以后每次一方取的个数不超过另一个方上次的两倍，不能取输
  - 结论：斐波那契数处先手必败，否则先手必胜
- 威佐夫博奕
  - 定义：两堆石子，每次任选一堆取走任意个，或者两堆同时取走同样多个，不能取者输
  - 结论：令两堆石子个数为$a,b(a\leq b)$，令$k=\frac{\sqrt{5}+1}{2}$，若$(int)k(b-a)=a$，则先手必败，否则先手必胜，即必败态$\{\lfloor k\cdot\frac{1+\sqrt{5}}{2}\rfloor, k+\lfloor k\cdot\frac{1+\sqrt{5}}{2}\rfloor\}(k=0,1,2,\cdots)$
  - 扩展：两堆石子，个数分别为$n,m(n\leq m)$，两人轮流从某一堆拿任意数量的物品或同时从两堆中取绝对值$\leq k$的物品，每次至少一个，不能操作的人败；$d=k+1, t^2+(d-2)*t-d=0$，解出t，$(m-n)/d*t=n$则为必败态
- 不平等博弈
  - $surreal\ number$基础
    - 定义：是一个包含无穷小、无穷大和实数的域，用它可以十分方便地研究一些不平等博弈问题。$surreal\ number$定义为一个集合对 $\{L|R\}$，其中 L 中的每个元素小于 R 中的每个元素。我们称 $x\leq y$ 当且仅当不存在 $x_L\in X_L$ 使得 $y\leq x_L$，且不存在 $y_R\in Y_R$ 使得 $y_R\leq x$。
    - 性质：
      - 若$x=\{L|R\}$，则$L<x<R$，表示x严格大于L中的任何一个元素，严格小于R中的任何一个元素
      - 令L_{max}表示L中最大值，则$x=\{L_{max}|R_{min}\}$；注意L,R可以为空，这时可以认为其最值为正负无穷大，不影响结论
      - 若$a<x<b$，且x是在a和b之间最早出现的(在树结构中层数最小)，则可证明x是唯一的，且$x=\{a|b\}$
    - 运算：
      - 加法法则：$x+y=\{X_L|X_R\}+\{Y_L|Y_R\}=\{X_L+y, x+Y_L|X_R+y, x+Y_R\}$
      - 取反法则：$-x=-\{X_L|X_R\}=\{-X_R|-X_L\}$
      - 乘法法则：
        $xy=\{X_L|X_R\}\{Y_L|Y_R\}=\{X_Ly+xY_L-X_LY_L,\ X_Ry+xY_R-X_RY_R|X_Ly+xY_R-X_LY_R,\ xY_L+X_Ry-X_RY_L\}$
      - 求逆法则：$\frac{1}{y}=\{0, \frac{1+(y^R-y)(\frac{1}{y})^L}{y^R}, \frac{1+(y^L-y)(\frac{1}{y})^R}{y^L}|\frac{1+(y^L-y)(\frac{1}{y})^L}{y^L}, \frac{1+(y^R-y)(\frac{1}{y})^R}{y^R}\}$
  - 判定结论
    - 用一个$surreal\ number\ x=\{L|R\}$表示一个局面，$L$集合表示左边玩家的决策空间(子局面)，则可利用$x$的值分类判定游戏状态的胜负情况：
      - $x>0$，L玩家总是获胜，无论他是先手还是后手
      - $x<0$，R玩家总是获胜，无论他是先手还是后手
      - $x=0$，后手获胜

        - $x||0$，先手获胜；这种情况x超出了实数的表示范围是一类无穷类数，表示在0附近且和0不可比的实数外的数
    - 游戏和：直接求出每个子游戏的$surreal\ number$值x_i，则总游戏值为其和$\sum_i x_i$
    - 更多结论，请参阅参考文献
  - 一些常见的无穷类数的表示

$$
\begin{aligned}
&\{0|1,\frac{1}{2},\frac{1}{4},\cdots\}=\epsilon(\text{无穷小})\\
&\{0,1,2,3,\cdots|\}=\omega(\text{无穷大})\\
&\{\epsilon|1,\frac{1}{2},\frac{1}{4},\cdots\}=2\epsilon\\
&\{0|\epsilon\}=\frac{\epsilon}{2}\\
&\{0,1,2,3,\cdots,\omega|\}=\omega+1\\
&\{0,1,2,3,\cdots|\omega\}=\omega-1\\
&\{\omega|\omega+1\}=\omega+\frac{1}{2}
\end{aligned}
$$

- 纳什均衡理论
  - 纯策略均衡：画出$payoff$矩阵，容易检查某个状态是否是纯策略均衡，第一维在行中取最值，第二维在列中取最值，意思是当处于这个组合状态时，任何一方不会轻易改变自己的策略，这个组合是占优的
  - 混合策略均衡：纳什均衡点指的是一组策略，表示某一方对决策空间有一个概率分布，有一个固定的概率选择某个决策，双方均有；在此纳什均衡点处，双方任何一方，不会率先改变自己的策略，因为这样只会使得自己受益减少；无论对方选择何种具体决策，在此点处我们的期望受益$\sum_{c\in S}P_c*V_c$是相等的
  - 纳什均衡点存在定理：在允许混合策略的情况下，任意一个有限个玩家的，每个玩家有有限个可选纯策略的游戏，必然有至少一个纳什均衡点
  - 典型例子
    - 智猪博弈
- nim积理论
  - 定义：$x\otimes y=\text{mex}\{(a\otimes b)\oplus(a\otimes y)\oplus(x\otimes b),0\le a<x,0\le b<y\}$
  - 数表：

	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	3	1	8
3	0	3	1	2	12
4	0	4	8	12	6

  - 运算性质：
    - $$
      \begin{aligned}
      &x\otimes 0=0\otimes x=0\\
      &x\otimes 1=1\otimes x=x\\
      &x\otimes y=y\otimes x
      \end{aligned}
      $$
    - $$
      \begin{aligned}
      &x\otimes(y\otimes z)=(x\otimes y)\otimes z\\
      &x\otimes(y\oplus z)=(o\otimes y)\oplus(x\otimes z)
      \end{aligned}
      $$
    - $\forall n\in N_+,[0,2^{2^n}-1]$中的数在$(\otimes,\oplus)$下构成一域
    - $\forall n\in N_+,2^{2^n}\otimes x=2^{2^n}\times x(0\le x<2^{2^n})$
    - $\forall n\in N_+,2^{2^n}\otimes 2^{2^n}=\frac{3}{2}\cdot 2^{2^n}=2^{2^n}\oplus 2^{2^n-1}$
  - 复杂度：模板计算一次大约复杂度$O(log^2)$
  - 应用：高维硬币游戏
    - 考虑二维网格平面，选择一个正方形(边长≥ 2)，取角落4个格子，同时翻转硬币，要求右上角(横纵坐标都大)的一个格子是从正面翻到反面的，不能翻者败，其$sg(x,y)=x\otimes y$
    - 可以扩展成三维或者更高维，$sg(x,y,z)=x\otimes y\otimes z$
    - 考虑另一个版本的硬币游戏，每次选择一个正方形(边长≥ 2)，取右上角和左下角翻转，则$sg(x,y)=x\oplus y$
- $shannon$开关游戏
- 广义地理游戏
- $k-knights$游戏
  - 定义：棋盘上有k只马，每次玩家对每只马都要操作，每只马只能向左下角四个方向走(日子)，当无法操作者输

- 结论：由于是一种$Every$游戏，故单一游戏需要满足：能够胜利的尽可能慢的胜利，不能胜利尽可能快的输掉；令$round(x,y)$表示单一的马在(x,y)时最快的步数，结论为：$round(x,y)-\lfloor\frac{x}{2}\rfloor-\lfloor\frac{y}{2}\rfloor$有规律(除了左下角$4\times4$和最外围一圈，可以$dp$)。

【124】代数：奇数双阶乘对2^p分块模数列的性质

函数$f(n)=1\times3\times5\times\cdots\times n$，其中$n$为奇数，则令$b_n=f(2^n)\%2^p$。则$b_{kn}=b_k^n$对$k\geq\frac{p+1}{3}$恒成立。若已知$b_k=B$，则$f(n)=B^{\lfloor n/2^k\rfloor}*\prod_{k\in[\lfloor n/2^k\rfloor\times2^k+1,n]\text{且}\,odd(k)}k$，复杂度$O(2^k)$。

【125】代数：关于下降阶乘幂的多项式间的运算

- 两个下降阶乘幂多项式的乘法
  - 令$f(x)=\sum_{i=0}^{+\infty}a_ix^{\underline{i}},A(x)=\sum_{i=0}^{+\infty}a_ix^i$
  - $g(x)$与$f(x)$形式类似，要求$f(x)g(x)$一种思路是利用斯特林数将两个多项式本身的具体系数求出来，然后用fft/ntt做多项式乘法，但是那样没有利用好阶乘幂本身的性质
  - 不难发现：$\frac{k^{\underline{i}}}{k!}=\frac{1}{(k-i)!}$
  - 我们考虑绕过$f(x)$本身的具体系数，直接求其点值序列的指数型生成函数$F(x)$
  - $$F(x)=\sum_{k=0}^{+\infty}\frac{f(k)}{k!}x^k=\sum_{k=0}^{+\infty}\frac{x^k}{k!}\sum_{i=0}^{+\infty}a_ik^{\underline{i}}=\sum_{i=0}^{+\infty}a_i\sum_{k=0}^{+\infty}\frac{x^k}{(k-i)!}=e^x\sum_{i=0}^{+\infty}a_ix^i=e^xA(x)$$
  - 我们已知$A(x)$，即可从$F(x)$中取出点值序列，$G(x)$类似，点值序列直接点乘，构成指数型生成函数$F_{ret}(x)$，再利用上述公式逆向求得$A_{ret}(x)=e^{-x}F_{ret}(x)$，取出系数即可

【126】图论：图有关的定理

- $Petersen's\,theorem$
  - $3-$正则无桥边的无向图，必存在完美匹配(此图不可能有奇数个点)
  - 推广：$d-$正则无向图，边连通度$\geq d-1$(至少删除$d-1$条边，图才可能不连通)，若点数为偶数，则必存在完美匹配；更进一步的，对每一条边，都存在一个包含此边的完美匹配
- $Tutte\,theorem$
  - 无向图$G=(V,E)$，存在完美匹配，当且仅当$\forall U\subseteq V$，$G-U$至多含有$|U|$个奇数联通块
  - 即$G=(V,E),\forall U\subseteq V,odd(G-U)\leq|U|$，其中$odd(G)$表示$G$中奇数连通块的个数
- $Tutte-Berge\,formula$
  - $G=(V,E)$的最大匹配的个数$=\frac{1}{2}min_{U\subseteq V}(|U|-odd(G-U)+|V|)$
- $Matrix\,Tree\,theorem$
- $BEST\,theorem$