博弈问题总结

目录

- 博弈问题总结
  - 组合游戏
    - 有向图的核
    - 经典模型
      - 巴什博弈
      - 威佐夫博弈
      - 多阶段博弈
  - SG 组合游戏
    - SG 函数
    - 游戏的和
    - NIM 游戏
      - NIM_k 游戏
  - SG 组合游戏的变形
    - anti-SG 游戏
    - multi-SG 游戏
      - Lasker's NIM 游戏
      - Octal 游戏
    - every-SG 游戏
  - 翻硬币游戏
    - 约束条件举例
      - Mock Turtles 游戏
      - Ruler 游戏
  - 无向图删边游戏
    - 树形删边游戏
    - 圣诞树删边游戏
    - 无向图删边游戏
  - 动态减法问题
    - 1 / 2 倍动态减法问题
    - k 倍动态减法问题
      - 动态规划法
      - 构造法
  - 高维组合游戏
    - NIM 积
      - 计算 NIM 积
      - Tartan 游戏
  - 其他方法举例

组合游戏

组合游戏

- 游戏有两个人参与，二者轮流做出决策。
- 在游戏中的任意一个时刻，当前玩家可以选择的操作，是一个有限的确定的集合。
- 在游戏中的任意时刻，游戏中的任意一方都知道游戏的初始状态，以及从“游戏开始”到“当前”为止的双方的所有的历史操作。

组合游戏 - 判断题
来猜一猜哪些是组合游戏吧！

- 2048
- 石头剪子布
- 斗地主
- 五子棋
- 中国象棋

组合游戏

- 我们需要研究的组合游戏：
- 游戏中的同一个状态不可能多次抵达，且游戏不会有平局出现。

组合游戏

- 将组合游戏中的每一个状态抽象成图中的一个点，将每一步决策抽象成图中的一条边。我们将这个有向图称为该组合游戏的游戏图。
- 游戏图的一个起始顶点上有一枚棋子，两名选手交替将这枚棋子沿有向边进行移动，无法移动者判负。这个游戏可以看作是所有公平组合游戏 (Impartial Combinatorial Games) 的抽象模型。
- 对于组合游戏中的每一次对弈，我们都可以将其抽象成游戏图中的一条从某一顶点到出度为 0 的点的路径。

有向图的核

- 对于有向图 G 的一个顶点集 S，若 S 是一个独立集并且 S 之外的每一个结点都能够经一步到达 S，则 S 被称为 G 的一个核。
- 定理：有限个结点的无回路有向图存在唯一的核。
- 定理：有向无环图的核唯一对应有向图游戏的全部必败态。
- 若先手面对的局面在核内且游戏仍未结束，那么他只能把棋子转移到核外，从而后手可使局面进入 核->非核->核->非核……->核 的循环。

有向图的核

- 关于有向图的核的唯一性的证明：
  - 无出度结点必然在核内。
  - 数学归纳：假设结点 x 的所有后继均为非核结点，则 x 在核中；否则 x 不在核中。
  - 从而我们可以构造出这个有向图的唯一的核。

热♂身的时间
POJ2505 A MULTIPLICATION GAME

- 从前有一个可♂爱的数字 p，绝顶聪明的 Stan 和 Ollie 想要让它变♂大。
- p 的初始值为 1，从 Stan 开始，两人轮流对数字 p 进行操作，每一次可以让 p 乘上 2 到 9 之间的一个数字。
- 第一个使得 p 大于 n (1 < n < 4294967295) 的人可以取得胜利。
- 最终胜利的会是谁呢?

热♂身的时间
POJ2505 A MULTIPLICATION GAME

- 我们首先来将数字抽象成游戏图内的结点。
- 当 p 大于 n 时，游戏结束，因此 (n, n*9) 为没有出边的结点。
- 逆推，[ceil((n+1)/9), n] 这些结点都存在指向核的出边，不属于核，
- [ceil(ceil((n+1)/9)/2), floor(n/9)] 这些结点的出边都在核外，都属于核……
- 这样，我们可以一步一步推出 p=1 是否属于有向图的核。

热♂身的时间
POJ2068 NIM

- 圆桌上有 s 个石子。喜欢看人们之间互相争斗的你将 2*n 个人分成了两伙，让他们交替坐在桌旁，顺时针轮流取石子，第 i 个人一次可以取 $[1,m_i]$ 个石子。
- 取到最后一个石子的一伙人要接受你的惩♂罚。现在问题来了……

热♂身的时间
POJ2068 NIM

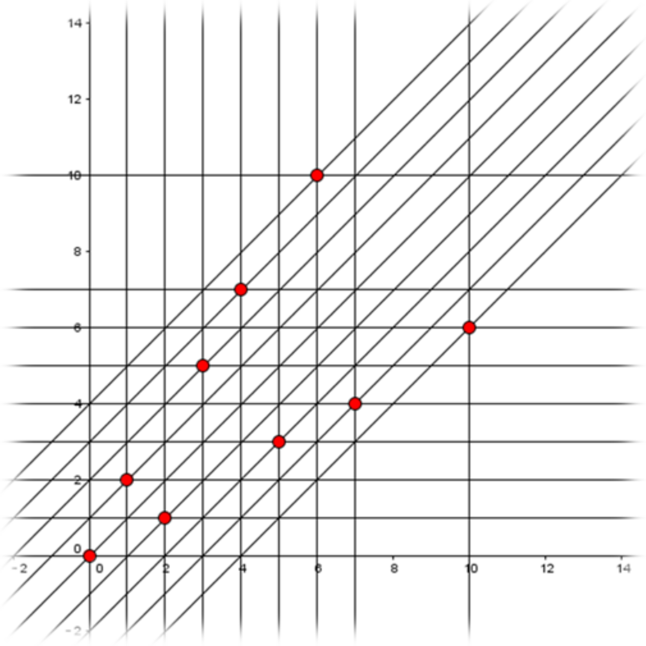
- 每一个状态都可用数对 (s,p) 表示，其中 s 表示桌面上剩余的石子个数，p 表示现在进行决策的人的编号。
- 显然，桌面上只剩一个石子的状态都对应着一个必败态，之后就可以利用 DP 进行转移了。

巴什博弈 (BASH GAME) POJ2368 BUTTONS

- 桌面上有 n 个按钮，要让两个人轮流将它们全部按下，其中按下最后一个按钮的人将会取得胜利。你可以指定一个非零整数 t，使得这两个人每一次按下的按钮数量在 [1,t] 之间。能确保第二个人失败的最小的 t 是多少呢?
- 显然，游戏的核为 $\{k*(t+1) \mid k \in N\}$，能够胜利的一方总能够使得在一回合内石子的数量减少 t+1 个。回到原题，答案就是 n 的最小非 1 约数

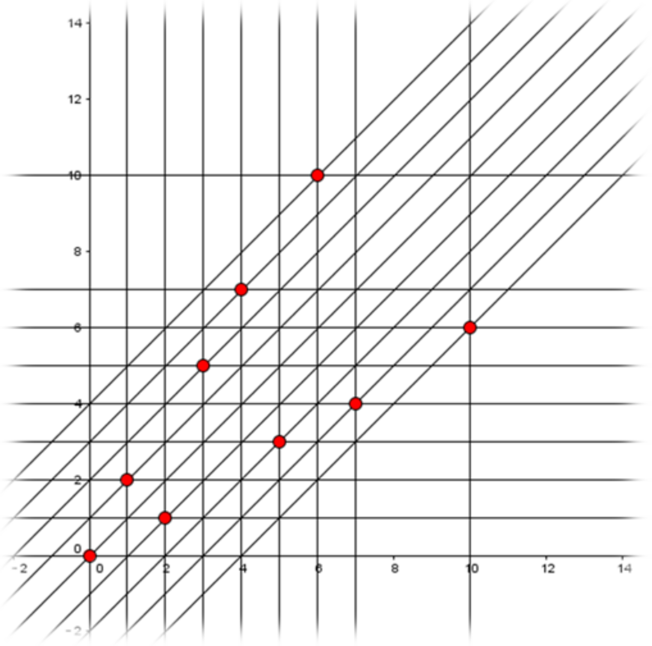
威佐夫博弈 (WYTHOFF GAME) POJ1067 取石子游戏

- 有两堆石子，数量任意，可以不同。游戏开始由两个人轮流取石子。游戏规定，每次有两种不同的取法，一是可以在任意的一堆中取走任意多的石子；二是可以在两堆中同时取走相同数量的石子。最后把石子全部取完者为胜者。现在给出初始的两堆石子的数目，如果轮到你先取，假设双方都采取最好的策略，问最后你是胜者还是败者。
- 显然如果一个点是核，那么它右面的、上面的和右上方的点就都不会是核了。打个表，如右图所示。
- 不妨设第一堆的数目较少，可以发现第 k 个必败态的两个数相差 k，且两个数都没有在之前的必败态中出现过。



威佐夫博弈 (WYTHOFF GAME) POJ1067 取石子游戏

- 从前有个神奇的定理，它叫做 Beatty 定理。
- 对于正无理数 α, β，若 $\frac{1}{\alpha}+\frac{1}{\beta}=1$，则数列$\{a_n\}: a_n = \lfloor \alpha n \rfloor, \{b_n\}: b_n = \lfloor \beta n \rfloor$ 严格递增，且每一个正整数在两个数列中出现且仅出现一次。
- 考虑另外一个限制条件，可以得到 $\alpha+1=\beta$。令 $\alpha = \frac{1+\sqrt{5}}{2}, \beta = \frac{3+\sqrt{5}}{2}$，发现恰好满足所有条件，于是就可以构造出第 k 个必败态为 $\left(\lfloor \frac{1+\sqrt{5}}{2} k \rfloor, \lfloor \frac{3+\sqrt{5}}{2} k \rfloor\right)$ 啦。



多阶段博弈 (MULTI-STAGE GAME) POJ2348 EUCLID'S GAME

- Stan 和 Ollie 这一次得到了两个数，他们要让这两个数变♂小。
- 他们从 Stan 开始轮流进行操作，每一次操作需要让较大的数减掉较小的数的倍数。谁能让其中一个数变成 0，就会获得胜利。胜利的会是谁呢?
- 不妨设这两个数为 p、q (p >= q)。当 p % q 为 0 时，先手将会获得胜利。否则两个人会轮流让 p 减去 q，直到 p’ = p % q，游戏进入下一个阶段，p 和 q 的位置互换。

多阶段博弈 (MULTI-STAGE GAME) POJ2348 EUCLID'S GAME

- 这个游戏等价于下面一个游戏：
- 有若干堆石子排成一排，两人轮流从最前面一堆取石子，每次至少取一个，只有当前面的石子全部取完时才能取后面堆的石子，最后取光者胜。
- 其中石子的堆数就是我们进行欧几里得算法（扣得一手好题）时递归的层数，石子的数量就是递归到每一层时 a / b 的取值。
- 我们来考虑上述游戏的必胜状态。若一个堆内石子个数为 1，则是否必胜取决于之后的游戏是否必胜。若一个堆的石子个数大于 1，那么刚取到这个堆时，先手可以自由选择转移到剩余石子数为 1 或直接开始之后的游戏，先手必胜。
- 我猜这道题大家早就会了..233

多余的小练习

- POJ1678 I Love this Game!
- HDU2177 取(2堆)石子游戏

SG 组合游戏

SG 函数

- SG 函数 (*Sprague-Grundy Function*) 是对游戏图中每一个节点的评估函数。
- $f(u) = \mathrm{mex}\{f(v) | \exists < u, v > \in E\}$.
- $\mathrm{mex}\{A\} = \min\{k | k \notin A \wedge k \in N\}$.
- 我们可以根据游戏图的拓扑关系来逐一算出每一个状态点的 SG 函数。

SG 函数

- SG 函数的性质：
  - 对于任意的局面，如果它的 SG 值为 0，那么它的任何一个后继局面的 SG 值不为 0；
  - 对于任意的局面，如果它的 SG 值不为 0，那么它一定有一个后继局面的 SG 值为 0。
- SG 组合游戏中，一个局面为先手必败的局面当且仅当该局面的 SG 函数值为 0。

游戏的和

- 考虑任意多个同时进行的 SG 组合游戏，这些 SG 组合游戏的和是这样一个 SG 组合游戏，在它进行的过程中，游戏者可以任意挑选其中的一个单一游戏进行决策，最终，没有办法进行决策的人输。
- SG 定理：游戏的和的 SG 函数值是所有子游戏的 SG 函数值的异或和。
- 这一条性质可用归纳法简单证明。

NIM 游戏

- 桌子上有 N 堆石子，游戏者轮流取石子。
- 每次只能从一堆中取出任意数目的石子，但不能不取。
- 取走最后一个石子者胜。
- Bouton’s Theorem：局面为先手必败当且仅当各堆石子数的异或和为 0。
- 对于只有一堆石子的取石子游戏，它的 SG 函数值与石子的个数相等。从而 NIM 游戏的 SG 函数值就是各堆石子个数的异或和。

NIM_K 游戏

- 如果每次可以同时从 k 堆中取至少一枚石子呢?
- 将石子数转换成二进制数后在 k+1 进制下进行不进位加法即可。

几道傻题

- POJ2234 Matches Game
- POJ2975 Nim
- POJ1704 Georgia and Bob
- POJ2960 S-Nim
- POJ2425 A Chess Game
- 到此为止，一切的一切的题目的画风都是如此地正常。

SG 组合游戏的变形

ANTI-SG 组合游戏

- anti-SG 组合游戏规定，决策集合为空的游戏者赢。
- anti-SG 组合游戏的其他规则与 SG 游戏相同。

ANTI-SG 组合游戏

- misère 规则尼姆游戏 (anti-NIM 游戏)
  - 取走最后一个石子者败，其他规则同 NIM 游戏。
- 对于一个 misère 规则尼姆游戏，先手必胜当且仅当
  - 所有堆的石子数都为 1 且游戏的 SG 值为 0；
  - 存在某个堆的石子数大于 1 且游戏的 SG 值不为 0。
- 由于这个结论用到了“单一游戏的 SG 值为 0 的局面一定为终止局面”这一性质，因此不能推广到所有的 anti-SG 组合游戏。

ANTI-SG 组合游戏

- 然而我们可以给出这一定理：
- 对于任意一个 anti-SG 游戏，如果我们规定当局面中所有的单一游戏的 SG 值均为 0 时，游戏结束，则先手必胜当且仅当
  - 游戏的 SG 函数不为 0 且游戏中某个单一游戏的 SG 函数大于 1；
  - 游戏的 SG 函数为 0 且游戏中没有单一游戏的 SG 函数大于 1。

ANTI-SG 组合游戏

- 证明：
  - 所有的终止局面为先手必胜局。
  - 情况一：局面的 SG 函数为 0 且游戏中某个单一游戏的 SG 函数大于 1 ⇒先手必败；
  - 情况二：局面的 SG 函数不为 0 且游戏中没有单一游戏的 SG 函数大于 1 ⇒先手必败；
  - 情况三：局面的 SG 函数不为 0 且游戏中某个单一游戏的 SG 函数大于 1 ⇒先手必胜；
  - 情况四：局面的 SG 函数为 0 且游戏中没有单一游戏的 SG 函数大于 1 ⇒先手必胜。

MULTI-SG 组合游戏

- 在符合拓扑原则的前提下，一个单一游戏的后继可以为多个单一游戏。
- 仅仅是多了几种后继状态而已，并没有什么特别的。

LASKER'S NIM 游戏

- 取石子游戏，每一次操作可以选择进行下列两个操作之一：
  - 从某一堆中取出任意数目的石子；
  - 选出石子个数大于等于 2 的一堆，将其分成两堆。
- 这是一个经典的模型，我们考虑单一游戏的 SG 函数值：
  - SG(0) = 0，SG(1) = 1，SG(2) = 2，SG(3) = 4；
  - 对于 k >= 1，SG(4k) = 4k-1，SG(4k+1) = 4k+1，SG(4k+2) = 4k+2，SG(4k+3) = 4k+4。
- 最后把每一个单一游戏的 SG 函数值异或到一起就好了。

OCTAL 游戏 / DAWSON'S CHESS

- 取石子游戏，最开始只有一堆共 n 个石子，每一次可以选择进行下列两个操作之一：
  - 从某一堆中取出任意数目的石子；
  - 从某一堆中取出任 意数目的石子，将剩余石子分成两堆，其中每一堆都非空。

POJ3537 CROSSES AND CROSSES

- 两个人在 1*n 的格子里面轮流画 x。当某个人画完之后出现了三个连着的 x，那么这个人就赢了。最后谁会赢呢?
- 发现当一个格子被画上 x 之后，它左右两侧与它距离小于等于 2 的格子就都不能再被画 x 了。因此一次操作等价于删除连续的五个元素（边界除外），然后将原游戏划分成一或两个子游戏。
- 依次算出对于每一个 n 的 SG 函数即可，时间复杂度 $O(n^2)$。

BZOJ1457 棋盘游戏

- 一个边长为 100 的棋盘，有 n 个皇后，初始位置给你。如果一个皇后在 (x, y)，那么你能将她移动到 (x-k, y) 或 (x, y-k) 或 (x-k, y-k)。谁先将任意一个皇后移动到 (0, 0) 谁获胜。
- 这道题大家都做过吧。我们需要将点 (1, 2) 和点 (2, 1) 的 SG 函数值设为 0，而不能直接将原点的 SG 函数值设为 0。若初始时在边界上存在皇后，需要特判。这是为什么呢?

BZOJ1457 棋盘游戏

- 注意到胜利条件是将任意一个皇后移到 (0, 0)，而不是让所有皇后都移动到 (0, 0)。如果让原点作为终止状态，就可能有“存在皇后能够决策，但游戏已经结束了”的情况出现。

POJ2311 CUTTING GAME

- 给一张 n*m 的纸片，两个人轮流去剪，每一次可以挑一张纸剪成两半。
- 谁先剪出来了一张 1*1 的小纸片，谁就能获得胜利。
- （题意描述什么的，大家理解就好~）

POJ2311 CUTTING GAME

- 和棋盘问题一样，我们需要设纸片的边长都小于等于 3 为终止状态。
- 剪一次的含义就是将现有游戏分割成两个子游戏。

EVERY-SG 组合游戏

- 对于还没有结束的单一游戏，游戏者必须对该游戏进行一步决策。
- 游戏的过程，就是所有单一游戏一个一个结束的过程，游戏的胜负取决于最后一个结束的游戏的胜负。
- 对于自己必胜的游戏，我们当然会想让它尽量长地持续下去。而对自己必败的游戏，我们会想让它们越早结束越好。

EVERY-SG 组合游戏

- 对于 SG 值为 0 的点，我们需要知道最快几步能让游戏终止，而对于 SG 不为 0 的点，我们需要知道最慢几步会使游戏结束。让这个值用 step 函数表示。
- $step(u) = \begin{cases} 0 & u \text{ 为终止状态}, \\ \max\big(step(v)\big)+1 & SG(u) > 0 \text{ 且 } v \text{ 是 } u \text{ 的后继状态且 } SG(v)=0, \\ \min\big(step(v)\big)+1 & SG(u)=0 \text{ 且 } v \text{ 是 } u \text{ 的后继状态}. \end{cases}$

EVERY-SG 组合游戏

- 对于 every-SG 游戏，先手必胜当且仅当单一游戏中最大的 step 为奇数。
- 感性理解一下，正确性是不是很显然~

EVERY-SG 组合游戏

- 简单说明:
- 对于所有的单一游戏，先手必胜状态的 step 值为奇数，先手必败状态的 step 值为偶数（可用数学归纳法简单证明）；
- 设最大的 step 为 max _step ，那么胜手可以保证该单一游戏最少会在 max _step 步结束;
- 设最小的 step 为 min _step ，那么胜手可以保证所有他必败的游戏最多在 min _step 步结束（按照定义进行分析可证）。
- 在论文《组合游戏略述——浅谈SG游戏的若干拓展及变形》给出了详细证明。

几道小练习而已

- POJ3480 John
- HDU3032 Nim or not Nim?
- HDU4664 Triangulation
- BZOJ2940 [Poi2000]条纹
- HDU3595 GG and MM
- BZOJ1393 [Ceoi2008]knights【点我下载题面】
- 好像这里有一些题已经步入了奇怪的轨道啊。

翻硬币游戏

翻硬币游戏

- 一些硬币排成一列，有的正面朝上，有的反面朝上。
- 两人轮流翻硬币，每次可翻一些硬币，但他所翻动的最右侧硬币必须由正变反。操作可能带有限制，例如一次翻两枚？一次翻一段？每一次翻的硬币编号互质？只能翻面值为一元的硬币？
- 不能操作者输。
- 结论：局面的 SG 值为局面中每个正面朝上的棋子单一存在时的 SG 值的异或和。

翻硬币游戏

- 结论：局面的 SG 值为局面中每个正面朝上的棋子单一存在时的 SG 值的异或和。
- 例子：SG(●○●○●●○) = SG(●●●●●●○) xor SG(●●●○) xor SG(●○).
- 记 SG(x) 为左起第 x 个硬币正面朝上的子游戏的 SG 函数值。

翻硬币游戏

- 简单说明：
- 考虑一次决策，可以认为我们在最右边的改动位去掉了一个正面朝上的硬币，在其他改动位添加了一个正面朝上硬币。
- 若其他改动位本来就是正面朝上的，那么多一个正面朝上的硬币之后，异或一下也就相当于是将原来的硬币翻到反面去了，并不会影响答案。
- 这样，对于每一个正面朝上的硬币，我们都可以将它看作一个子游戏，且不同的子游戏之间互不影响。

一些约束条件举例

- 每一次只能翻转一枚硬币 ----- SG(0) = 0，SG(k) = 1 (k > 0)；
- 每一次可以翻转一枚或两枚硬币 ----- 相当于 NIM 游戏；
- Twins 游戏：每次必须翻动两个硬币，而且这两个硬币的距离要在可行集 S={1,2,3} 中 ----- 相当于巴什博弈；
- 每一次必须翻连续的 n 个硬币 ----- SG(n*k) = 1 (k>0)，其他 SG 函数值为 0。

MOCK TURTLES 游戏
HDU3537 DAIZHENYANG'S COIN

- 每一次可以翻 1~3 枚硬币。
- 打表找规律吧！这才是博弈的精髓！（不妨设硬币的编号从 0 开始）
- x 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 …
- SG(x) 1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 …
- 规律1：SG(x) ≈ 2x；
- 规律2：SG(x) 的二进制的 1 个数为奇数。

MOCK TURTLES 游戏
HDU3537 DAIZHENYANG'S COIN

- 简单说明：
- 称一个非负整数为 odious，当且仅当该数的二进制形式的 1 出现的次数是奇数，否则称其为 evil。
  - evil xor evil = odious xor odious = evil，evil xor odious = odious xor evil = odious。
- SG(x) = mex{SG(0), …, SG(x-1), SG(0)^SG(0), …, SG(x-1)^SG(x-1)}.
- 这些数中，前半段恰好覆盖了不大于 2x 的 odious 数，后半段覆盖了不大于 2x 的 evil 数。

RULER 游戏

- 每一次可以翻任意长度的连续一段硬币。
- SG(x) = mex(0, SG(x-1), …, SG(x-1) xor … xor SG(1)).

x	0	1	2	3	4	5	6	7	8	9
	10	11	12	13	14	15	16	…		
SG(x)	0	1	2	1	4	1	2	1	8	1
	2	1	4	1	2	1	16	…		

- SG(x) 为 x 中包含的 2 的最高次幂。

无向图删边游戏

树形删边游戏

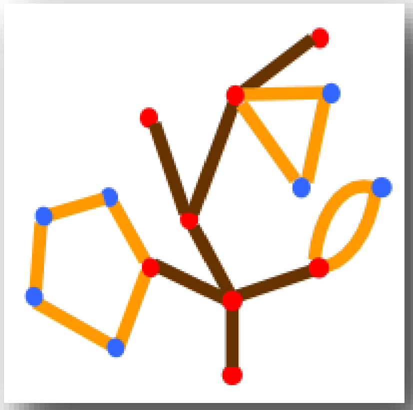
- 给出一棵有 n 个结点的有根树。两个人轮流砍树，之后把砍下的子树拿去卖钱。谁先没树可砍谁输。
- Colon 原理：SG(u) = XOR{SG(v)+1 | v 是 u 的子节点}。

树形删边游戏

- 简单说明：
- （数学归纳法）一个结点和两个节点的情况显然成立。
- 假设小于 k 个结点的情况均成立，证明 k个结点仍然成立，只需证：
  - 根节点只有一个孩子的情况成立（显然）；
  - 根节点有多个孩子的情况成立（分解成第一种情况）。

圣诞树删边游戏
POJ3710 CHRISTMAS GAME

- 在树上挂几个环，每一个环只与树有一个公共点，环与环之间无公共边。
- 对于长度为奇数的环，去掉一条边后得到两条长度同奇偶的链，SG值异或后不可能是奇数，因此SG值为1
- 对于长度为偶数的环，去掉一条边后可以得到两条长度异奇偶的链，SG值异或后不可能是偶数，因SG值为0。



无向图删边游戏

- 将有根树变成一个有根无向连通图呢？
- Fusion 原理：环上的点可以被融合，且不改变图的 SG 值。
- 推论：我们可以对无向图做如下改动，将图中的任意一个偶环缩成一个新点，任意一个奇环缩成一个新点加一个新边；所有连到原先环上的边全部改为与新点相连。这样的改动不会影响图的 SG 值。
- 这个原理的证明十分复杂。
- 实现？其实只需要求边双连通分量缩点就可以了。

动态减法游戏

K 倍动态减法游戏

- 有一个整数 S，两个人想让它变成 0。首先，第一个人需要把 S 减掉一个数 x ($0 < x < S$)。之后双方轮流把 S 减掉一个正整数，但都不能超过先前一回合对方减掉的数的 k 倍，减到 0 的一方获胜。问谁会获得胜利。

1 倍动态减法游戏

- 有一个整数 S，两个人想让它变成 0。首先，第一个人需要把 S 减掉一个数 x ($0 < x < S$)。之后双方轮流把 S 减掉一个正整数，但都不能超过先前一回合对方减掉的数，减到 0 的一方获胜。问谁会获得胜利。
- 结论：当且仅当 S 是 2 的幂次时，后手必胜。

2 倍动态减法游戏

- 有一个整数 S，两个人想让它变成 0。首先，第一个人需要把 S 减掉一个数 x ($0 < x < S$)。之后双方轮流把 S 减掉一个正整数，但都不能超过先前一回合对方减掉的数的 2 倍，减到 0 的一方获胜。问谁会获得胜利。
- 结论：当且仅当 S 是一个斐波那契数时，后手必胜。
- 当 k <= 2 时，我们对于 k 倍动态减法问题均可在 O(logS) 内给出答案。

2 倍动态减法游戏 BZOJ2275 [COCI2010]HRPA

- 如果先手要有必胜策略，第一次他至少要取多少个。
- 将 s 减剩一个最大的斐波那契数？显然这是不对的，因为后手并不是随意取数，这样操作不能保证答案最小。
- 齐肯多夫定理：任何正整数可以唯一的表示为若干个不连续的斐波那契数之和。
- 这一分解的过程可以贪心地每一次分解出最大的斐波那契数。

齐肯多夫定理

- 存在性:
- 假设对1-m-1都成立，考虑m，若m是斐波那契数，显然成立。否则令n1为满足f(n1)<m<f(n1+1)的正整数，那么t=m-f(n1)<f(n1+1)-f(n1)=f(n1-1)，且t已经被证明可以被分解，故其分解出的最大的斐波那契数必定小于f(n1-1)

齐肯多夫定理

- 唯一性:
- 设n为最小的有两种分解方法的正整数，设其中一种分解出的最大数为a，另一种为b。
- 引理：若m为斐波那契数，则从小于等于m的斐波那契数中选取任意多个不相邻的数，能够成的最大的数小于m的下一个斐波那契数。
- 例：8:8+3+1=12<13

齐肯多夫定理

- 所以若a!=b，不妨令a>b,则b这组不可能构造出大于等于a的数。
- 故a=b, n-a=n-b
- 由假设，n-a也有至少两种分解方法，与n为最小的有至少两种方法分解的正整数矛盾。

2 倍动态减法游戏 BZOJ2275 [COCI2010]HRPA

- 我们将 s 分解之后，可以得到一个类似于多阶段博弈的模型。
- 先手可以依次减光分解出的小斐波那契数，并且保证每一个小斐波那契数都由自己最终减光。由于斐波那契数都是不连续的，相邻两个小数字之差一定大于其中较小的数字，所以可以保证后手不会一下子减掉一个小斐波那契数。虽然我不会证明为什么后手不会一下子减掉小的那个数先手就能赢，但是感受一下大概挺有道理的2333
- 总之这样先手就可以保证胜利啦！
- 于是我们得到了答案即为从 s 中分解出的最小的斐波那契数。

K 倍动态减法游戏

- 利用单调性在关于 S 的线性时间内解决。
- 用数对 (m, n) 表示 S 还剩下 m，当前回合操作者最多可以减去 n 的状态。
- 设函数 NP(m, n) 表示状态 (m, n) 的必胜状态，即若 (m, n) 状态先手必胜，NP(m, n) = 1，否则 NP(m, n) = 0。
- 显然，函数 NP 的取值关于 n 单调不减，即若 NP(m, n) = 1，对于所有的正整数 r > n，NP(m, r) = 1。

K 倍动态减法游戏

- 设函数 $f(m)$ 表示使得 $NP(m, n) = 1$ 的最小的 n。由于 $NP(m, m)$ 总是等于 1 的，因此 $f(m) <= m$。这里规定 $f(0)$ 为正无穷，因 $NP(0, n)$ 的取值总会是 0。
- 我们知道，$NP(m, n) = 0$ 当且仅当 (m, n) 的所有后继状态的 NP 值为 1，即当且仅当对于任意正整数 $r <= n$，有 $m > r$ 且 $NP(m-r, kr) = 1$。
- 设 $f(m) = n_0$，那么我们就可以得到 $n_0=1$ 或 $NP(m-n_0+1, k(n_0-1)) = 1$ 必然满足其一，并且 $NP(m-n_0, kn_0) = 0$，即 $f(m-n_0+1) <= k(n_0-1)$ $(n_0 > 1)$ 且 $f(m-n_0) > kn_0$。
- 于是我们只需要从小到大依次枚举 m，逐个检验对于每一个 n 是否满足上述式子即可。时间复杂度 $O(S^2)$。

K 倍动态减法游戏

- $f(m-n_0+1) <= k(n_0-1)$ $(n_0 > 1)$ 且 $f(m-n_0) > kn_0 \Rightarrow f(m) = \min\{n \mid f(m-n) > kn\}$。
- 进一步，我们可以发现，$f(m)$ 就是从 $(m, 0)$ 格子出发，斜率为 k^{-1} 的直线遇到第一个紫色点走过的距离。
- 注：右图中表示的是 $k=1$ 的情况。

```
0
0 0
0
0 0 0 0
0
0 0
0
0 0 0 0 0 0 0 0
```

K 倍动态减法游戏

- 一层一层的紫色点像墙一样摞在一起。
- 发现所有的斜线都会是平行的，因此随着 m 的逐渐增大，若一堵墙已经不能再遮挡住斜线，那么它以后也不会再遮挡住斜线了。
- 我们可以用栈来维护这些墙，时间复杂度便能够优化到 O(S) 了。

```
0
0 0
0
0 0 0 0
0
0 0
0
0 0 0 0 0 0 0 0
```

K 倍动态减法游戏

- 求出来函数 f 之后，我们还可以在 O(1) 的时间里得到某个状态的必胜走法。在状态 (m, n) 时，当前玩家将 S 减去 f(m) 就是一个必胜策略，并且可以证明这也是减去数字最少的策略。

K 倍动态减法游戏

- 到这里，并没有结束。事实上还有在一些情况下更优越的算法！
- 还记得 $k = 2$ 时的斐波那契数列么？
- 这个数列满足性质：任意一个正整数都可以拆分成斐波那契数列中几项之和，并且这几项在数列中两两之间不相邻。
- 之后，我们利用斐波那契数列不相邻的两个数之间，较大的数大于较小的数的二倍的性质，构造了斐波那契博弈的必胜方案。

K 倍动态减法游戏

- 当 2 扩大到 k 的时候，我们也可以类似地构造一个数列。例如当 k = 3 时，数列长成下面的样子：
- a_n = 1, 2, 3, 4, 6, 8, 11, 15, 21, ...
- 这个数列同样满足每一个正整数都可以拆分成序列中元素之和，并且拆分出的每两个元素间较大的数大于较小的数的 k 倍。
- 我们对于每一个 k 都可以通过一种神奇的方式构造出来一个序列。

K 倍动态减法游戏

- 首先，这个数列的首项为 1。
- 假设我们已经构造出了序列的前 i 项 a[1]...a[i]，它们从 1 开始能够构造出的最大连续数字为 b[i]，那么 a[i+1] = b[i] + 1。
- 然后我们找到最大的 t 使得 a[t] * k < a[i+1]，b[i+1] 的值即为 a[i+1]+b[t]。
- t 是随着 i 的递增而递增的，感受到优越的复杂度的气息了么?

裸题时间

- HDU2486 A simple stone game
- HDU2580 a simple stone game
- POJ3922 A simple stone game

高维组合游戏

翻角游戏
HDU3404 SWITCH LIGHTS

- 把硬币排列成一个矩阵，左上角从 (0, 0) 开始编号。
- 两个人轮流进行翻硬币的操作，每次操作选择两个硬币 (x, y)、(x’, y’)，其中硬币 (x, y) 正面朝上，并且 x’ < x， y’ < y。之后需要翻动 (x, y)、(x’, y)、(x, y’)、(x’, y’) 四个硬币，不能操作者输。
- 问最后的胜利情况。

翻角游戏

- 我们仍然可以考虑只有一枚正面朝上的硬币时的情况。
- $SG(0, x) = SG(x, 0) = 0$,
- $SG(x, y) = \mathrm{mex}\{SG(x', y') \text{ xor } SG(x, y') \text{ xor } SG(x', y)\}$.

翻角游戏

- 然后就打了个表。
- 记这个游戏的 SG 函数 $SG(x, y) = x \otimes y$。于是我们就得到了 NIM 乘法运算。
- $x \otimes y = \text{mex}\{(a \otimes b) \text{ xor } (x \otimes b) \text{ xor } (a \otimes y)\}$.
- $0 <= a < x, 0 <= b < y$.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	0	2	3	1	8	10	11	9	12	14	15	13	4	6	7	5
3	0	3	1	2	12	15	13	14	4	7	5	6	8	11	9	10
4	0	4	8	12	6	2	14	10	11	15	3	7	13	9	5	1
5	0	5	10	15	2	7	8	13	3	6	9	12	1	4	11	14
6	0	6	11	13	14	8	5	3	7	1	12	10	9	15	2	4
7	0	7	9	14	10	13	3	4	15	8	6	1	5	2	12	11
8	0	8	12	4	11	3	7	15	13	5	1	9	6	14	10	2
9	0	9	14	7	15	6	1	8	5	12	11	2	10	3	4	13
10	0	10	15	5	3	9	12	6	1	11	14	4	2	8	13	7
11	0	11	13	6	7	12	10	1	9	2	4	15	14	5	3	8
12	0	12	4	8	13	1	9	5	6	10	2	14	11	7	15	3
13	0	13	6	11	9	4	15	2	14	3	8	5	7	10	1	12
14	0	14	7	9	5	11	2	12	10	4	13	3	15	1	8	6
15	0	15	5	10	1	14	4	11	2	13	7	8	3	12	6	9

TARTAN 游戏

- 考虑两个一维的翻硬币游戏 G_1 和 G_2，我们可以根据这两个游戏来构造一个 Tartan 游戏 $G_1 \times G_2$：
- 如果在 G_1 的某个位置我们可以同时翻转硬币 $x_1, x_2, \ldots, x_m$，在 G_2 的某个位置我们可以同时翻转硬币 $y_1, y_2, \ldots, y_n$，那么在二维游戏 $G_1 \times G_2$ 中的对应位置上， 可以同时翻转硬币 (x_i, y_j)，其中 $i <= m$，$j <= n$。

TARTAN 定理

- 如果游戏 G_1 的 SG 函数为 $g_1(x)$，游戏 G_2 的 SG 函数为 $g_2(x)$，
- 那么二维游戏 $G_1 \times G_2$ 的 SG 函数 $g(x, y) = g_1(x) \otimes g_2(y)$ 。

NIM 积的运算法则

- 存在单位元和逆元，即 x 不为 0 时 $x \otimes 1 = 1 \otimes x = x$，存在 y 使得 $x \otimes y = 1$。
- 满足交换律、结合律以及对 NIM 和（就是异或）的分配律。
- 存在 $O(\log^2 x)$ 的算法，能够让我们快速求出两个数的 NIM 积。

NIM 积的性质

- NIM 积的性质：
- 对于 x, y < 2^{2^a}，设 t = 2^{2^a}：

1. x⊗y < t;
2. x⊗t = x * t;
3. t⊗t = (3 / 2) * t * t。

计算 NIM 积

- 计算 X⊗Y（不妨设 X > Y）：
- 令 X = p*M + q，Y = s*M + t，其中 M=$2^{2^{a}}$，$2^{2^{a}}$ <= X < $2^{2^{a+1}}$，q, t < M。
- 将上式中的 “*” 变成 “⊗”，将 “+” 变成异或。
- 利用分配律，X⊗Y = (p⊗M)⊗(s⊗M) ^ q⊗(s⊗M) ^ (p⊗M)⊗t ^ q⊗t。
- 利用结合律，X⊗Y = (p⊗s)⊗(M⊗M) ^ (q⊗s ^ p⊗t)⊗M ^ q⊗t。
- 利用NIM积的性质 (p⊗s)⊗(M⊗M) = (p⊗s)⊗M ^ (p⊗s)⊗(M / 2)。

计算 NIM 积

```
int nim_multi(int x, int y)
{
    if (x < y) swap(x, y);
    if (y == 0) return 0;
    if (x == 1) return 1;
    int t = 2; while (t * t <= x) t = t * t;
    int c1 = nim_multi(x / t, y / t),
        c2 = nim_multi(x / t, y % t) ^ nim_multi(x % t, y / t),
        c3 = nim_multi(x % t, y % t);
    return (c1 ^ c2) * t ^ c3 ^ nim_multi_power(t / 2, c1);
}
```

计算 NIM 积

- 这样，我们通过一系列变换缩减了 X、Y 的规模，使它们都小于 2^{2^a}。
- 我们每一层要递归去计算4个没什么规律的 NIM 积和一个M/2和c1的NIM积，f(n)=4f(n-1)+g(n-1),g(n)=3g(n-1)
- 最后算完可以得到f(n)=4^n，其中n为层数，为loglog max(x,y)
- 这样，我们的最坏时间复杂度就是 $O(4^{\log\log X})$，约等于$O(\log^2 X)$。

RUGS 游戏

- 上面的翻角游戏可看作是两个 NIM 游戏的乘积。
- Rugs 游戏同样给出硬币矩阵，然而我们要翻动的不止是四角，而是矩阵内部的所有硬币。
- 可看成是两个 Ruler 游戏的乘积。

一模一样的另一道题

- POJ3533 Light Switching Game

喜闻乐见的例题时间
用奇怪的方式解决博弈论问题

海盗分金问题
HDU1538 A PUZZLE FOR PIRATES

- 有 n 个海盗来分 m 块金子。
- 海盗的编号是从 1 到 n。从第 n 号海盗开始，他们会依次提出自己的分配方案。如果半数及以上的人(包括半数）赞成他的方案，则方案通过，开始分金子；否则方案不通过，海盗们就会把提出方案的那个人分尸吃掉，由下一个人继续提出方案。

海盗分金问题
HDU1538 A PUZZLE FOR PIRATES

- 每一个海盗在提出方案和投票时，都遵循着这样一个原则，即：
- 先保证自己的生命安全，再想办法得到最多的金子，最后还要努力吃掉更多的人。
- 由于海盗很懒并且都患有强迫症，所以金子不能分割。
- 现在要问的是对于每一个海盗，他能否活下去？如果能的话，他至少可能分到多少金子？

海盗分金问题
HDU1538 A PUZZLE FOR PIRATES

- 其实这与其说是博弈题，不如说是推理题。
- 从头开始推理吧！如果只有一个海盗，那么金子都是他的。
- 如果有两个海盗，那么第二个海盗的方案就是把所有金子都给自己，自己投赞成票。第一个海盗虽然很不满，但是无可奈何。
- 如果有多个海盗，那么就开始变得有趣起来了。

海盗分金问题
HDU1538 A PUZZLE FOR PIRATES

- 有三个海盗。如果他把金子都留给自己，那么仇恨一定会害死自己的。
- 如果他给了第二个人金子，那么第二个人不会领情，因为如果第三个人死了的话所有的金子都是他一个人的。
- 如果他不给第一个人金子，那么第一个人会想：反正我分不到金子，不如把你吃掉；然而哪怕一块金子，都能获得第一个人的支持。
- 于是对第三个人而言的最优分法就是给第一个人一块金子，给自己剩余金子，然后得到第一个人和自己的赞成票。

海盗分金问题
HDU1538 A PUZZLE FOR PIRATES

- 之后的情况类似，第 n 个人分金子的时候，会给编号与自己同奇偶的人一人一块金子，剩余金子归自己所有。

海盗分金问题
HDU1538 A PUZZLE FOR PIRATES

- 然而这个世界上和谐的事情不总是会那么多。如果金子不够分了怎么办?
- 有 m 块金子，当第 2m+1 个人在分金子的时候，他自己就已经分不到任何金子了。
- 推想之后的情况，第 2m+2 个人在分金子时实际上就已经捉襟见肘了，而第 2m+3 个人就真正无能为力了。他无论把金子给谁，都会有 m+2 个人反对。那么，他必死无疑。

海盗分金问题 HDU1538 A PUZZLE FOR PIRATES

- 第 2m+4 个人在分金子的时候，第 2m+3 个人无论如何不会想让他死，因为他死完之后自己也要陪他去。所以，第 2m+4 个人凭空多出来一张赞成票，再加上那 m 块金子换来的票和自己的票，刚刚好活命。
- 之后，第 2m+5 个人必死；第 2m+6 个人得到了第 2m+5 个人的票，可惜还不够，必死；第 2m+7 个人得到了第 2m+5 个人和第 2m+6 个人的两票，然而还是去死了；第 2m+8 个人得到了上面三个人的三票，刚刚好活了下来。类似地，第 $2m+2^k$ 个人其实都是可以保证自己活下来的。

海盗分金问题
HDU1538 A PUZZLE FOR PIRATES

- 接下来我们分析每一个人的收益。n <= 2m+1 的情况，我们之前已经讨论过了。当 n >= 2m+2 时，我们假设没用的人已经全被吃掉了，现在轮到第 $2m+2^k$ 个人公布方案。
- 显然，编号大于 $2m+2^{k-1}$ 的人是没有人权的，他们为了活命只能投赞成票，并且绝对不会得到金子。
- 我们接下来用归纳法来证明其他人都不一定能获得金子。

海盗分金问题
HDU1538 A PUZZLE FOR PIRATES

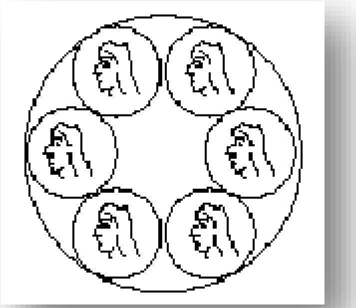
- 当 n = 2m+2 时，如果送给编号为 2m+1 的人一块金子，他也会开心地投赞成票。这时，可以发现没有人可以确保自己得到金子了。
- 对于 n > 2m+2 的每个方案，由于大家都不确定自己在下一个人的分配方案中能否得到金子，因此如果能够得到金子的话，都会高兴地投赞成票。这时，奇偶性的限制已经不存在了，无论什么人（当然，没有人权的人不算人）只要得到金子就会投赞成票，否则就一定会投反对票。
- 总之题目问的是“至少”，所以当 n >= 2m+2 时，所有人的答案都是 0。

海盗分金问题
HDU1538 A PUZZLE FOR PIRATES

- 这个故事告诉我们，剿灭海盗最好的方式，就是在海盗的聚集区放合适数量的金子。

对称性思想
POJ2484 A FUNNY GAME

- 我猜我并不需要翻译这个题目。
- 我猜我并不需要讲这个题目。



对称性思想
JDFZOJ2586 THE ULTIMATE BATTLE

- 一个战场被分割成 W*H 的网格，游戏双方开始时，站在两个不同行也不同列的的格子里。双方轮流操作。每个玩家每次操作，可以水平的或垂直的移动若干格，不能走出战场。另外，两个人手中都是有枪的，而只有双方在同一水平线上，或同一垂直线上时，才能开枪。因此，“移动”操作不能穿过对方所在的水平线或垂直线。问谁有必胜策略？

对称性思想
JDFZOJ2586 THE ULTIMATE BATTLE

- 我们第一次移动之后要使得两人之间横、纵坐标的差值相等。
- 如果对方沿横（纵）坐标倒退，只需沿横（纵）坐标前进同样的格子；
- 如果对方沿横（纵）坐标前进，只需沿纵（横）坐标前进同样的格子。
- 这样，当对方移动一次之后，我们总可以移动相应的一步，直到把对方逼到角落。

REFERENCE

- 《eolv-博弈问题总结》

谢谢大家