Combinatorial Game Theory

Chunyang Wang

2020 年 12 月 20 日

Combinatorial Game Theory

- 组合博弈(Combinatorial Game Theory, CGT)是数学和理论计算机科学的一个分支，一般研究信息公开的顺序(sequential)博弈。

Combinatorial Game Theory

- 组合博弈(Combinatorial Game Theory, CGT)是数学和理论计算机科学的一个分支，一般研究信息公开的顺序(sequential)博弈。
- 简单的例子:Nim, Toads and Frogs, Hackenbush

Combinatorial Game Theory

- 组合博弈(Combinatorial Game Theory, CGT)是数学和理论计算机科学的一个分支，一般研究信息公开的顺序(sequential)博弈。
- 简单的例子:Nim, Toads and Frogs, Hackenbush
- 复杂一些的例子: Chess, Go

Combinatorial Game Theory

- 组合博弈(Combinatorial Game Theory, CGT)是数学和理论计算机科学的一个分支，一般研究信息公开的顺序(sequential)博弈。
- 简单的例子:Nim, Toads and Frogs, Hackenbush
- 复杂一些的例子: Chess, Go
- 并非组合博弈的例子: Hearthstone, DOTA2, Cyberpunk2077

Impartial and Partizan games

- 组合博弈一般来说分为两种:无偏博弈(Impartial game)和有偏博弈(Partizan game).

Impartial and Partizan games

- 组合博弈一般来说分为两种:无偏博弈(Impartial game)和有偏博弈(Partizan game).
- 无偏博弈指可允许的操作只和当前局面的状态有关而和操作的玩家无关的博弈，有偏博弈中可允许的操作则还和当前操作玩家相关。

Impartial and Partizan games

- 组合博弈一般来说分为两种:无偏博弈(Impartial game)和有偏博弈(Partizan game).
- 无偏博弈指可允许的操作只和当前局面的状态有关而和操作的玩家无关的博弈，有偏博弈中可允许的操作则还和当前操作玩家相关。
- 我们今天主要讨论无偏博弈。

An example for patizan game: Blue-red Hackenbush

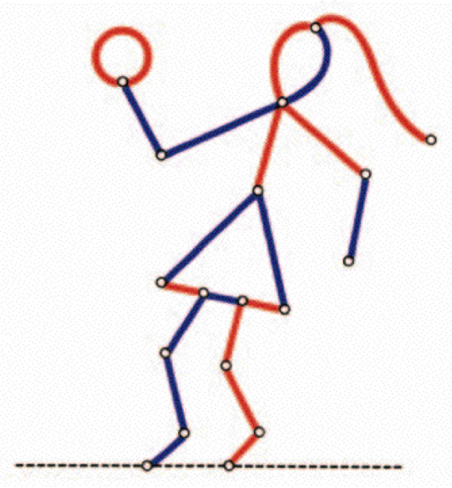

图: Who'll win this game?

Another example: Blue-red-green Hackenbush

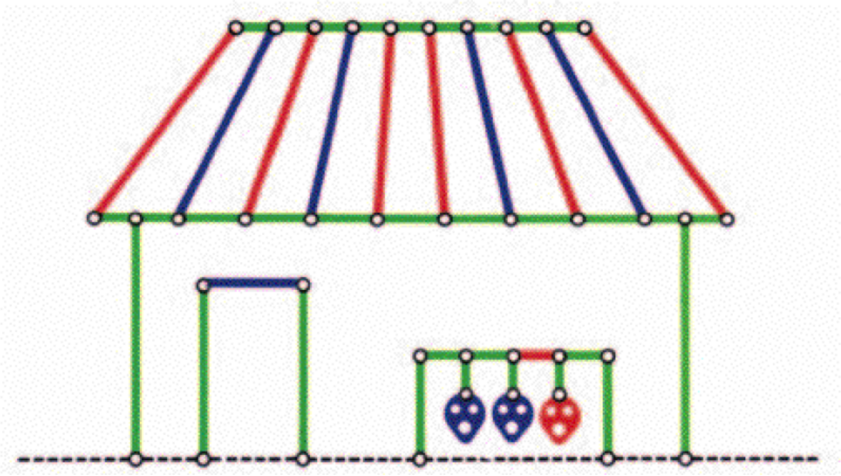

图: And how about this one?

Zermelo’s Theorem

策梅洛定理(Zermelo’s Theorem)

若一个游戏满足如下条件：

1. 双人、回合制
2. 信息完全公开（perfect information）
3. 无随机因素（deterministic）
4. 必然在有限步内结束
5. 没有平局

则游戏中的任何一个状态，要么先手有必胜策略，要么后手有必胜策略。

$\mathcal{P}$-positions and $\mathcal{N}$-positions

- 假设在双方最优策略下,如果在当前状态下行动的一方最终会获胜，我们就把当前状态称作必胜态($\mathcal{P}$-position)

$\mathcal{P}$-positions and $\mathcal{N}$-positions

- 假设在双方最优策略下,如果在当前状态下行动的一方最终会获胜，我们就把当前状态称作必胜态($\mathcal{P}$-position)
- 反之，如果在当前状态下行动的一方最终会失败，我们就把当前状态称作必败态($\mathcal{N}$-position)

$\mathcal{P}$-positions and $\mathcal{N}$-positions

- 假设在双方最优策略下,如果在当前状态下行动的一方最终会获胜，我们就把当前状态称作必胜态($\mathcal{P}$-position)
- 反之，如果在当前状态下行动的一方最终会失败，我们就把当前状态称作必败态($\mathcal{N}$-position)
- 在某些博弈中，可能会出现在某些状态中双方在最优策略下游戏无法结束的情况，这时我们把这种状态称为循环态(loopy position)

巴什博弈(Bash Game)

一堆石子有n个，两个玩家轮流拿石子，规定每次至少取一个，最多取m个，最后取光者得胜。

巴什博弈(Bash Game)

一堆石子有n个，两个玩家轮流拿石子，规定每次至少取一个，最多取m个，最后取光者得胜。

结论

如果$(m+1) \mid n$,那么后手获胜，否则先手获胜。

巴什博弈(Bash Game)

一堆石子有n个，两个玩家轮流拿石子，规定每次至少取一个，最多取m个，最后取光者得胜。

结论

如果$(m+1) \mid n$,那么后手获胜，否则先手获胜。

证明思路(通用)

1. 证明游戏结束时所在的状态和必胜/必败态定义相符合。
2. 证明对于所有必胜态总有一种方法能转移到必败态
3. 证明对于所有必败态只能转移到必胜态

威佐夫博弈(Wythoff Game)

有两堆石子，分别有n个和m个$(n \leq m)$,两个玩家轮流从任一堆取至少一个或同时从两堆中取同样多的石子，规定每次至少取一个，多者不限，最后取光者得胜。

威佐夫博弈(Wythoff Game)

有两堆石子，分别有n个和m个$(n \leq m)$,两个玩家轮流从任一堆取至少一个或同时从两堆中取同样多的石子，规定每次至少取一个，多者不限，最后取光者得胜。

结论

所有必败态满足$n_k = \lfloor k\phi \rfloor, m_k = \lfloor k\phi^2 \rfloor, k \geq 0$,其中$\phi = \frac{1+\sqrt{5}}{2}$是黄金比例。

威佐夫博弈(Wythoff Game)

有两堆石子，分别有n个和m个$(n \leq m)$,两个玩家轮流从任一堆取至少一个或同时从两堆中取同样多的石子，规定每次至少取一个，多者不限，最后取光者得胜。

结论

所有必败态满足$n_k = \lfloor k\phi \rfloor, m_k = \lfloor k\phi^2 \rfloor, k \geq 0$,其中$\phi = \frac{1+\sqrt{5}}{2}$是黄金比例。

- 证明/推导需要了解贝蒂定理(Beatty's theorem)

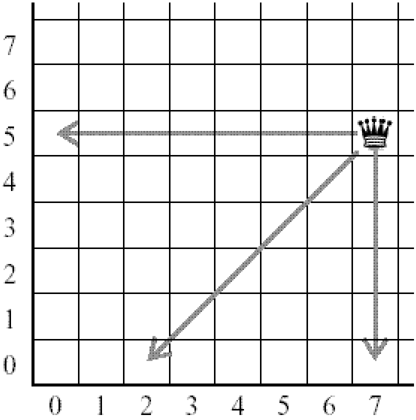

图: An analog of Wythoff's Game-Wyt Queens

尼姆博弈(Nim Game)

有n堆石子，数量分别是$a_1, a_2, a_3, \ldots, a_n$,两个玩家轮流选择任意一堆拿石子，规定每次至少取一个，多者不限，最后取光者得胜。

尼姆博弈(Nim Game)

有n堆石子，数量分别是$a_1, a_2, a_3, \ldots, a_n$,两个玩家轮流选择任意一堆拿石子，规定每次至少取一个，多者不限，最后取光者得胜。

结论

所有必败态满足$\bigoplus_{i=1}^{n} a_i = 0$,其中$\oplus$表示异或运算。

尼姆博弈(Nim Game)

有n堆石子，数量分别是$a_1, a_2, a_3, \ldots, a_n$,两个玩家轮流选择任意一堆拿石子，规定每次至少取一个，多者不限，最后取光者得胜。

结论

所有必败态满足$\bigoplus\limits_{i=1}^{n} a_i = 0$,其中$\oplus$表示异或运算。

引理

如果$a \oplus b < a$, 那么b为1的最高位上a也为1.

Sprague-Grundy Theorem

定义(mex函数)

对于一个定义域在非负整数上的集合S,它的mex函数定义为最小的不出现在集合S中的数，即$mex(S) = \min\{x|x \notin S\}$.

Sprague-Grundy定理(Sprague-Grundy Theorem)

任何一个一定会结束的无偏博弈的状态A等价于尼姆游戏中的一堆石子，我们把这堆石子的个数称作这个状态A的SG值(nimber),记作$SG(A)$。一个状态的nimber等于于所有其后继状态nimber的mex函数值,即:

1. SG值为0的状态，后手必胜。SG值不为0的状态，先手必胜。
2. 对于一个状态A，$SG(A) = mex\{SG(B)|A \to B\}$.
3. 若一个母状态可以拆分成多个相互独立的子状态，则母状态的SG值等于各个子状态的SG值的异或。

Wall Making Game

- 在一个$H \times W$的棋盘上，棋盘上的每个格子有空、被标记或者是墙三种状态。两个玩家轮流行动，每轮一个玩家可以选择一个空的格子，然后朝四个方向延伸，把所有经过的格子都变成墙(包括选中的格子)，直到碰到墙或者是棋盘的边界为止。谁先不能操作谁输，试判断最优策略下先手获胜还是后手获胜。
- Constraints: $1 \leq H, W \leq 20$
- Source: Petrozavodsk Summer-2015. JAG Contest, Problem L

Wall Making Game

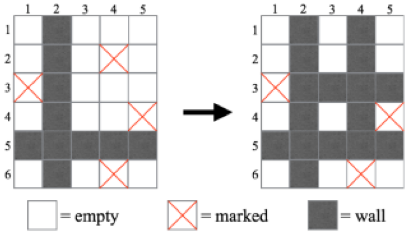

图: An example of a move in Wall Making Game

Wall Making Game

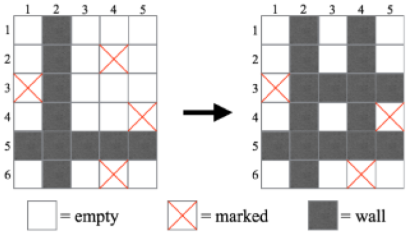

图: An example of a move in Wall Making Game

- DP计算每个子矩阵的SG值即可，时间复杂度能过。

Poker Nim

Poker Nim

有n堆石子，数量分别是$a_1, a_2, a_3, \ldots, a_n$,初始时两个玩家手上都没有石子，两个玩家轮流执行以下两种操作之一:

- 选择任意一堆非空石子，从中拿走任意正整数数量的石子。
- 选择任意一堆石子，从自己已经拿走的石子里分出任意正整数数量添加进去。

首先不能操作的玩家输。

Poker Nim

Poker Nim

有n堆石子，数量分别是$a_1, a_2, a_3, \ldots, a_n$,初始时两个玩家手上都没有石子，两个玩家轮流执行以下两种操作之一:

- 选择任意一堆非空石子，从中拿走任意正整数数量的石子。
- 选择任意一堆石子，从自己已经拿走的石子里分出任意正整数数量添加进去。

首先不能操作的玩家输。

结论

Poker Nim的胜败态和Nim Game相同。

Reversible Moves

定义(可逆操作)

如果一个状态A能转移到B，然后B也能转移到A,那么我们就把A转移到B这一操作称为可逆操作(reversible move)

Reversible Moves

定义(可逆操作)

如果一个状态A能转移到B，然后B也能转移到A,那么我们就把A转移到B这一操作称为可逆操作(reversible move)

Reversible Move Theorem

对于两个状态A和B，如果B可以通过A添加可逆操作得到，那么状态A和状态B等价。

The Silver Dollar Game

The Silver Dollar Game

在一个$1 \times n$的格子上放着一些硬币，初始时每个格子至多有一枚硬币。两个玩家轮流操作，一次操作可以选择一枚硬币并将它左移任意格，但不能越过其它硬币或与其它硬币重叠，首先不能操作的玩家输。

The Silver Dollar Game

The Silver Dollar Game

在一个$1 \times n$的格子上放着一些硬币，初始时每个格子至多有一枚硬币。两个玩家轮流操作，一次操作可以选择一枚硬币并将它左移任意格，但不能越过其它硬币或与其它硬币重叠，首先不能操作的玩家输。

- Poker Nim in disguise!

The Silver Dollar Game

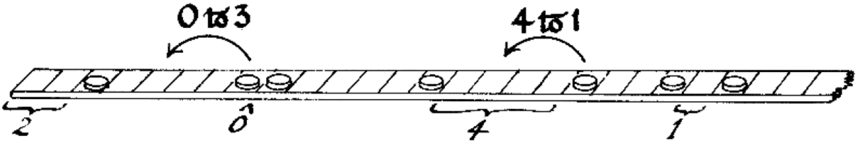

图: The Silver Dollar Game

Conway

- 客厅里有M个灯泡和N个开关，每个开关控制一些灯泡的集合，第i个灯泡的亮度是p_i，开始时所有灯泡都是不亮的。两个玩家Alice和Bob轮流进行$\frac{N-1}{2}$次操作，每轮都是Alice先操作。在第i轮，Alice可以选择按动编号$2i-1$和$2i$的开关，然后Bob可以选择按动编号$2i$和$2i+1$的开关。给定一个阈值K，如果最终房间的亮度大于等于K，那么Alice获胜，否则Bob获胜，试判断最优策略下是谁获胜。
- Constraints: $1 \leq N \leq 33, 1 \leq M \leq 32, 0 \leq K \leq 2 \cdot 10^9$, N是奇数。
- Source: Petrozavodsk Winter-2015. Moscow SU Tapirs Contest

Conway

- 除了最后一步，可以看做在第i轮，Alice可以选择按动编号$2i-1$的开关，然后Bob可以选择按动编号$2i$的开关。(为什么?)

Conway

- 除了最后一步，可以看做在第i轮，Alice可以选择按动编号$2i-1$的开关，然后Bob可以选择按动编号$2i$的开关。(为什么?)
- AND-OR Game Tree Evaluation!

Conway

- 除了最后一步，可以看做在第i轮，Alice可以选择按动编号$2i-1$的开关，然后Bob可以选择按动编号$2i$的开关。(为什么?)
- AND-OR Game Tree Evaluation!
- 直接搜索的话复杂度是$O(4^{\frac{n}{2}})$的

Conway

- 除了最后一步，可以看做在第i轮，Alice可以选择按动编号$2i-1$的开关，然后Bob可以选择按动编号$2i$的开关。(为什么?)
- AND-OR Game Tree Evaluation!
- 直接搜索的话复杂度是$O(4^{\frac{n}{2}})$的
- 如果每次随机check一个孩子的值可以用归纳法证明期望复杂度是$O(3^{\frac{n}{2}})$的。

Chomp

Chomp

有一个$n \times m$块的巧克力板，双方轮流选择一块巧克力，吃掉它及其右下方的所有巧克力，吃掉左上方的巧克力的玩家失败。

Chomp

Chomp

有一个$n \times m$块的巧克力板，双方轮流选择一块巧克力，吃掉它及其右下方的所有巧克力，吃掉左上方的巧克力的玩家失败。

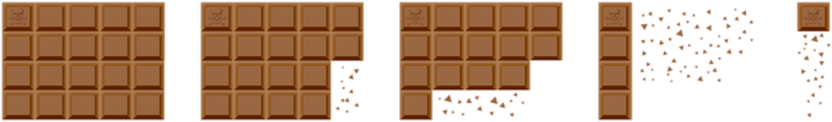

图: An example of a sequence of moves in Chomp

Chomp

Chomp

有一个$n \times m$块的巧克力板，双方轮流选择一块巧克力，吃掉它及其右下方的所有巧克力，吃掉左上方的巧克力的玩家失败。

结论

当$n = m = 1$的情况下先手必败，其他情况下先手必胜。

Chomp

Chomp

有一个$n \times m$块的巧克力板，双方轮流选择一块巧克力，吃掉它及其右下方的所有巧克力，吃掉左上方的巧克力的玩家失败。

结论

当$n = m = 1$的情况下先手必败，其他情况下先手必胜。

证明

Strategy Stealing!

MUV LUV UNLIMITED

- 在一棵有n个顶点的树上，两个玩家轮流删去至少一个叶子节点，不能操作的玩家输，试判断最优策略下先手获胜还是后手获胜。
- Constraints: $1 \leq n \leq 10^6$
- Source: 2019 China Collegiate Programming Contest Qinhuangdao Onsite, Problem K

- 令一个叶子节点一直往上到分叉点为止（不包括分叉点）的这一条链称为一个枝条

MUV LUV UNLIMITED

- 令一个叶子节点一直往上到分叉点为止（不包括分叉点）的这一条链称为一个枝条
- 如果某棵树上存在一个长度为1的枝条，那么先手必胜。(为什么?)

MUV LUV UNLIMITED

- 令一个叶子节点一直往上到分叉点为止（不包括分叉点）的这一条链称为一个枝条
- 如果某棵树上存在一个长度为1的枝条，那么先手必胜。(为什么?)
- 如果存在长度为奇数的枝条，那么先手必胜，否则后手必胜。

Green Hackenbush

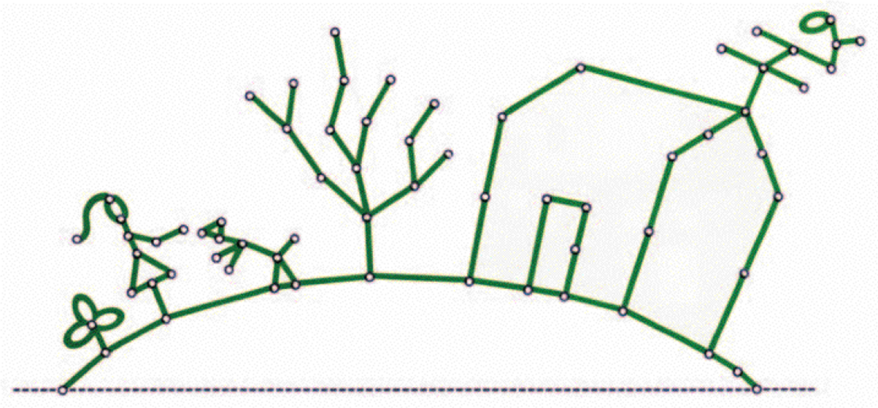

图: A Green Hackenbush Bridge

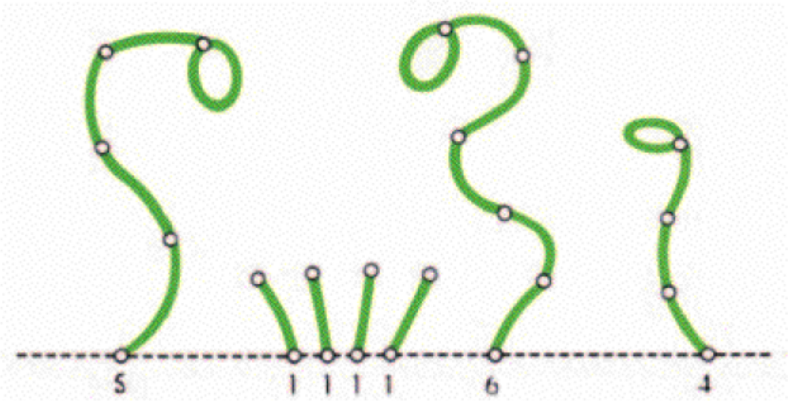

图: Green Snakes

- 这些Green Hackenbush的SG值是什么?

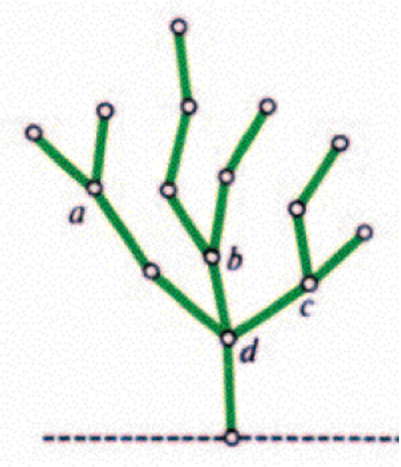

图: Green Trees

克朗原理(Colon's Principle)

对于树上的某一个点，它的分支可以转化成以这个点为根的一条链，这条链的长度就是它各个分支的边的数量的异或和

融合原理(The Fusion Principle)

环上的点可以融合，且不改变图的SG值。

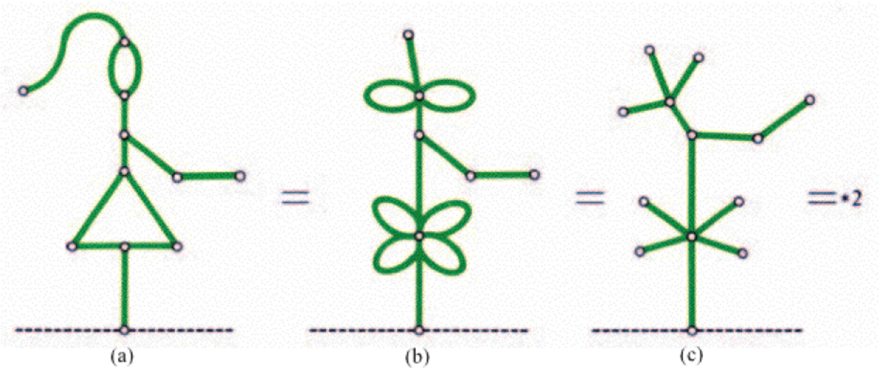

图: Sizing Up a Green Girl

EndGame

- 国际象棋中，黑方剩一个国王，白方剩一个国王和一个车先手。保证至多50步白方可以将军，黑方想最大化步数，白方想最小化步数，求白方多少步能将军。
- Source: Petrozavodsk Winter-2017. Jagiellonian U Contest, Problem D

EndGame

- 国际象棋中，黑方剩一个国王，白方剩一个国王和一个车先手。保证至多50步白方可以将军，黑方想最大化步数，白方想最小化步数，求白方多少步能将军。
- Source: Petrozavodsk Winter-2017. Jagiellonian U Contest, Problem D
- 更一般的问题：有向图+终止状态集合

EndGame

- 国际象棋中，黑方剩一个国王，白方剩一个国王和一个车先手。保证至多50步白方可以将军，黑方想最大化步数，白方想最小化步数，求白方多少步能将军。
- Source: Petrozavodsk Winter-2017. Jagiellonian U Contest, Problem D
- 更一般的问题：有向图+终止状态集合
- Similar Problems: 2014西安现场赛H、2016沈阳网络赛1006

EndGame

- 首先求出每个点的胜败态，win、lose点的胜败态是显然的，未确定点的状态按下列方法推：
  1. 如果一个未确定点有一个后继是必败的，那么它就是必胜的
  2. 如果一个未确定点后继全是必胜的，那么它就是必败的
  3. 无法继续更新时，剩下的未确定点就是最优策略下一定会走成环的平局状态

EndGame

- 首先求出每个点的胜败态，win、lose点的胜败态是显然的，未确定点的状态按下列方法推：
  1. 如果一个未确定点有一个后继是必败的，那么它就是必胜的
  2. 如果一个未确定点后继全是必胜的，那么它就是必败的
  3. 无法继续更新时，剩下的未确定点就是最优策略下一定会走成环的平局状态
- 接下来考虑步数，由于平局状态的步数都是正无穷所以可以把平局状态的点都去掉，然而剩下图还可能有环，第一想法可能会考虑设$step_{win} = step_{lose} = 0$,用$minstep = minstep + 1$ 和 $step = maxstep + 1$ 迭代

EndGame

- 首先求出每个点的胜败态，win、lose点的胜败态是显然的，未确定点的状态按下列方法推：
  1. 如果一个未确定点有一个后继是必败的，那么它就是必胜的
  2. 如果一个未确定点后继全是必胜的，那么它就是必败的
  3. 无法继续更新时，剩下的未确定点就是最优策略下一定会走成环的平局状态
- 接下来考虑步数，由于平局状态的步数都是正无穷所以可以把平局状态的点都去掉，然而剩下图还可能有环，第一想法可能会考虑设$step_{win} = step_{lose} = 0$,用$minstep = minstep + 1$ 和 $step = maxstep + 1$ 迭代
- 然而实际上观察(1)(2) 和迭代式发现只要保证按照队列扩展在满足(1)(2)条件时直接确定步数入队即可，时间复杂度$O(V + E)$

Games on DAG

- 给定一个N个点M条边的有向图，其中第i条边从x_i指向$y_i(x_i < y_i)$，且保证没有重边.
- 你现在可以选择M条边的一个子集并删除其中的所有边，然后Alice和Bob会在这个图上玩一个游戏:开始时在1号点和2号点上有两个token，Alice和Bob轮流操作，Alice先操作，每次将其中一个token沿一条有向边移动，谁先不能操作谁输。
- 问在所有2^M种删边的方案中，有多少种会使得Alice最终获胜，答案对$10^9 + 7$取模。
- Constraint:$2 \leq N \leq 15, 1 \leq M \leq \frac{N(N-1)}{2}$
- Source: AtCoder Grand Contest 016, Problem F

Games on DAG

- Alice能够获胜等价于1号点和2号点的SG值相同。

Games on DAG

- Alice能够获胜等价于1号点和2号点的SG值相同。
- 考虑fix一个序列$(g_1, g_2, \ldots, g_N)$,我们如何统计有多少个图满足第i个点的SG值是g_i?

Games on DAG

- Alice能够获胜等价于1号点和2号点的SG值相同。
- 考虑fix一个序列$(g_1, g_2, \ldots, g_N)$,我们如何统计有多少个图满足第i个点的SG值是g_i?
- 令S_i表示SG值为i的顶点集合，那么$g_i = x$等价于以下条件:
  1. 对于$y = 0, 1, 2, \ldots, x-1$,至少存在一条v到S_y的边。
  2. 不存在v到S_x的边。

Games on DAG

- Alice能够获胜等价于1号点和2号点的SG值相同。
- 考虑fix一个序列$(g_1, g_2, \ldots, g_N)$,我们如何统计有多少个图满足第i个点的SG值是g_i?
- 令S_i表示SG值为i的顶点集合，那么$g_i = x$等价于以下条件:
  1. 对于$y = 0, 1, 2, \ldots, x-1$,至少存在一条v到S_y的边。
  2. 不存在v到S_x的边。
- 枚举划分，可以得到一个$O(B_N \times poly(N))$的做法

Games on DAG

- 令$dp[S]$表示考虑只添加顶点集合S内部的边并使得1号点和2号点的SG值相同的方案数。

Games on DAG

- 令$dp[S]$表示考虑只添加顶点集合S内部的边并使得1号点和2号点的SG值相同的方案数。
- 考虑如何转移，我们枚举SG值为0的集合$U \subseteq S$，并令$T = S \backslash U$.,则应当满足以下条件.

Games on DAG

- 令$dp[S]$表示考虑只添加顶点集合S内部的边并使得1号点和2号点的SG值相同的方案数。
- 考虑如何转移，我们枚举SG值为0的集合$U \subseteq S$，并令$T = S \backslash U$.,则应当满足以下条件.
  1. T要么同时包含1和2，要么都不包含。
  2. U内部不应当有边。
  3. U向T的边可以任意添加
  4. 对于每个T中的顶点，至少要向U连一条边。
  5. T中连边的方案数恰好是$dp[T]$!

Games on DAG

- 令$dp[S]$表示考虑只添加顶点集合S内部的边并使得1号点和2号点的SG值相同的方案数。
- 考虑如何转移，我们枚举SG值为0的集合$U \subseteq S$，并令$T = S \backslash U$.,则应当满足以下条件.
  1. T要么同时包含1和2，要么都不包含。
  2. U内部不应当有边。
  3. U向T的边可以任意添加
  4. 对于每个T中的顶点，至少要向U连一条边。
  5. T中连边的方案数恰好是$dp[T]$!
- 预处理后可以做到$O(3^N \times N)$的复杂度。

Winning Ways For Your Mathematical Plays

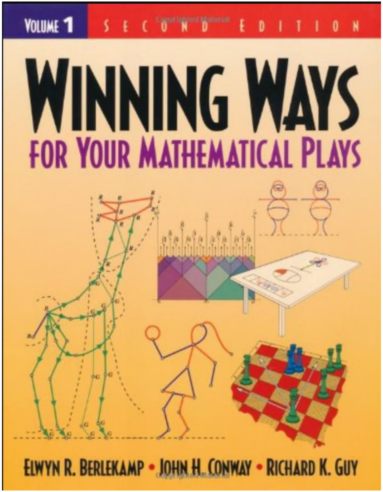

图: Winning Ways For Your Mathematical Plays. Vol 1.

Topics not covered

- Misère Nim and other Misère games
- Turning Games and Nim Multiplication
- Fibonacci Nim, Smallest/Largest Nim, Moore's Nim...
- Loopy Games and Smith Theorem
- Much, much more for partizan games...