博弈论滑铲入门

CraZYali

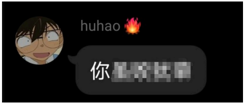
2020 年 10 月 26 日

说到底还是

~~水量巨大~~
这个滑铲入门部分仅仅介绍一些简单而常见的博弈。相关的描述可能也存在不够准确的地方，但不影响理解。
其他的 dirty works 一律丢给 IjfcnyaIi。

说到底还是

~~水量巨大~~
这个滑铲入门部分仅仅介绍一些简单而常见的博弈。相关的描述可能也存在不够准确的地方，但不影响理解。
其他的 dirty works 一律丢给 IjfcnyaIi。

没听懂？没关系，只要听完这节课：

不懂的欢迎找 X503-(D07/D08/C06) 一对一辅导。

Impartial Combinatorial Games

公平组合游戏，简称 ICG。OI 中**大部分**的博弈论题目都是这种游戏。

ICG

游戏有 2 人参与，轮流操作。参与者共享所有信息。
任意时刻任意一个参与者能做出的决策只与当前状态有关，与决策者**无关**。
状态有限，同一个状态不可多次到达，以非平局状态结束。

Impartial Combinatorial Games

公平组合游戏，简称 ICG。OI 中**大部分**的博弈论题目都是这种游戏。

ICG

游戏有 2 人参与，轮流操作。参与者共享所有信息。
任意时刻任意一个参与者能做出的决策只与当前状态有关，与决策者**无关**。
状态有限，同一个状态不可多次到达，以非平局状态结束。

有些题目可能出现平局，但没有必要拘泥于 ICG 的定义。

SG 游戏 SG 函数

一类组合游戏，无法操作者输。
游戏图 (V, E)：如果状态 u 能转移到状态 v，那么在游戏图中 u 向 v 连一条有向边。

SG 游戏 SG 函数

一类组合游戏，无法操作者输。
游戏图 (V, E)：如果状态 u 能转移到状态 v，那么在游戏图中 u 向 v 连一条有向边。SG 函数是对图中每一个状态的评估函数。

SG 函数

$$SG(u) = \mathrm{mex}\{SG(v) | (u, v) \in E\}$$

其中，$\mathrm{mex}(S) = \min\{x | x \in N, x \notin S\}$。根据定义，先手必败态的 SG 为 0。

游戏的和 Multi-SG

如果一个游戏 G 能被划分为若干个互不相关的游戏 $G_1, G_2 \cdots G_k$，那么称 G 为 $G_1, G_2 \cdots G_k$ 的和。
一个 Multi-SG 游戏满足单一游戏的后继是若干个单一游戏，其他规则与 SG 游戏相同。
此时 G 的一个局面的 SG 函数为 $G_1, G_2 \cdots G_k$ 的 SG 函数的**异或**和。

游戏的和 Multi-SG

如果一个游戏 G 能被划分为若干个互不相关的游戏 $G_1, G_2 \cdots G_k$，那么称 G 为 $G_1, G_2 \cdots G_k$ 的和。
一个 Multi-SG 游戏满足单一游戏的后继是若干个单一游戏，其他规则与 SG 游戏相同。
此时 G 的一个局面的 SG 函数为 $G_1, G_2 \cdots G_k$ 的 SG 函数的**异或**和。

证明无了（不会，做题时也没有必要）。

Nim - Statement

n 堆桃子，第 i 堆有 a_i 个。
CraZYali 和 cnyali_czy 轮流吃桃子。每人每次可从某一堆中吃掉任意多个桃子，可以吃完但不能不吃。
某一时刻，要吃桃子的人没有桃子可吃就输了。询问先手是否必胜。

Nim - Solution

显然每一堆桃子是独立的，求出每一堆桃子的 SG 然后异或起来即可。
定义 $f(x)$ 为一堆 x 个桃子的 SG 值，那么我们有：

定理

$$f(0) = 0$$
$$f(x) = \mathrm{mex}\{f(0), f(1), f(2) \cdots f(x-1)\}$$

Nim - Solution

显然每一堆桃子是独立的，求出每一堆桃子的 SG 然后异或起来即可。
定义 $f(x)$ 为一堆 x 个桃子的 SG 值，那么我们有：

定理

$$f(0) = 0$$
$$f(x) = \operatorname{mex}\{f(0), f(1), f(2) \cdots f(x-1)\}$$

显然的，我们发现 $f(x) = x$。
于是先手必胜当且仅当 $a_1 \operatorname{xor} a_2 \operatorname{xor} \cdots \operatorname{xor} a_n \neq 0$。

分裂游戏 - Statement(I'm a link!)

n 个瓶子，第 i 个初始有 a_i 个巧克力豆。
两人轮流操作。每次先选取三个数 $i < j \le k$，然后从 i 号瓶子中取出一个，再向 j, k 号瓶子各放一个（$j = k$ 也要放第二个）。
无法操作者输，问先手是否必胜。
$n \le 21, a_i \le 10^4$。

分裂游戏 - Statement(I'm a link!)

n 个瓶子，第 i 个初始有 a_i 个巧克力豆。
两人轮流操作。每次先选取三个数 $i < j \leq k$，然后从 i 号瓶子中取出一个，再向 j, k 号瓶子各放一个（$j = k$ 也要放第二个）。
无法操作者输，问先手是否必胜。
$n \leq 21, a_i \leq 10^4$。
实际上 a_i 开到 1145141919810 也没关系。~~n 可能也没关系（待考证）。~~

分裂游戏 - Solution

有一点脑筋急转弯的是，此题切入点是每个巧克力豆独立。
每次相当于移动一个巧克力豆，然后再加入一个巧克力豆。
发现由于后手可以模仿先手操作，a_i 可以对 2 取模。也可以从 SG 是异或得来理解。
直接 $\mathcal{O}(n^3)$ 计算 SG 就可以了。

分裂游戏 - Solution

有一点脑筋急转弯的是，此题切入点是每个巧克力豆独立。
每次相当于移动一个巧克力豆，然后再加入一个巧克力豆。
发现由于后手可以模仿先手操作，a_i 可以对 2 取模。也可以从 SG 是异或得来理解。
直接 $\mathcal{O}(n^3)$ 计算 SG 就可以了。
实际上，这个 SG 差分之后（至少项数比较小时）呈现一个 $1, 2, 3, 1, 2, 3 \cdots$ 的形式，不知道有没有人能证明一下。

无来源题 - Statement

有两个长度分别为 n, m 的整数序列 a, b 和两个分别指向 a, b 中位置的光标 c, d，保证 a, b 严格不增。一个局面的得分为 $a_c + b_d$。
两个人轮流操作。每次操作可以：

- 立即结束整个游戏并且计算得分。
- 否则，选取 c, d 中的某一个并且任意移动。
- 移动后的 (c, d) 状态不能在之前的时刻出现过。

先手想让得分尽量小，后手想让得分尽量大。求最终得分。
$n, m \leq 10^5$

无来源题 - Solution

二分答案，上界不会超过初始的 $a_c + b_d$。
为了方便描述，把 (c, d) 状态放在 $n \times m$ 的棋盘上。称 $a_c + b_d \le mid$ 的格子为白格，否则为黑格。
初始状态在黑格上。先手想最终停在白格上，后手反之。

无来源题 - Solution

二分答案，上界不会超过初始的 $a_c + b_d$。
为了方便描述，把 (c, d) 状态放在 $n \times m$ 的棋盘上。称 $a_c + b_d \leq mid$ 的格子为白格，否则为黑格。
初始状态在黑格上。先手想最终停在白格上，后手反之。注意到一次移动只可能移动到不同的颜色，否则对手会立即结束游戏。

无来源题 - Solution

二分答案，上界不会超过初始的 $a_c + b_d$。
为了方便描述，把 (c, d) 状态放在 $n \times m$ 的棋盘上。称 $a_c + b_d \le mid$ 的格子为白格，否则为黑格。
初始状态在黑格上。先手想最终停在白格上，后手反之。注意到一次移动只可能移动到不同的颜色，否则对手会立即结束游戏。
我们在同一行或同一列且颜色不同的格子之间连边，那么会成为一个**二分图，这是一个二分图博弈。**

这波怎么输?

定理

先手必胜当且仅当任何一个最大匹配方案都包含初始状态。

这波怎么输?

定理

先手必胜当且仅当任何一个最大匹配方案都包含初始状态。

证明.

充分性:
每次先手都从匹配边走到对面，而后手走回来的边一定可以不是匹配边，那么先手又可以用匹配边走过去。

这波怎么输?

定理

先手必胜当且仅当任何一个最大匹配方案都包含初始状态。

证明.

充分性:
每次先手都从匹配边走到对面,而后手走回来的边一定可以不是匹配边,那么先手又可以用匹配边走过去。
必要性:
如果存在一个最大匹配方案不包含初始节点,那么对面节点在此方案中一定都被匹配,进而在所有方案中都被匹配。

这波怎么输?

定理

先手必胜当且仅当任何一个最大匹配方案都包含初始状态。

证明.

充分性:
每次先手都从匹配边走到对面，而后手走回来的边一定可以不是匹配边，那么先手又可以用匹配边走过去。
必要性:
如果存在一个最大匹配方案不包含初始节点，那么对面节点在此方案中一定都被匹配，进而在所有方案中都被匹配。
此时后手变成了先手，且初始节点一定在最大匹配上，后手必然获胜。 □

这波怎么输?

定理

先手必胜当且仅当任何一个最大匹配方案都包含初始状态。

证明.

充分性:
每次先手都从匹配边走到对面,而后手走回来的边一定可以不是匹配边,那么先手又可以用匹配边走过去。
必要性:
如果存在一个最大匹配方案不包含初始节点,那么对面节点在此方案中一定都被匹配,进而在所有方案中都被匹配。
此时后手变成了先手,且初始节点一定在最大匹配上,后手必然获胜。 □

可我怎么知道初始状态是不是恒在最大匹配上啊?

如何判定?

最暴力的方法：删了这个初始节点，再跑一边匹配，如果最大匹配不变那他就是不必要的。

如何判定?

最暴力的方法：删了这个初始节点，再跑一边匹配，如果最大匹配不变那他就是不必要的。
巧妙一点的：从初始节点开始遍历所有点，如果能找到一个同侧的非匹配点就可以一路替换上去——那么初始节点就是不必要的，否则就是必要的。

无来源题 - Solution

考虑 最大匹配 $=$ 总点数 $-$ 最大独立集，那么删去初始点后最大匹配变化等价于最大独立集变化。

无来源题 - Solution

考虑 最大匹配 $=$ 总点数 $-$ 最大独立集，那么删去初始点后最大匹配变化等价于最大独立集变化。
记 r_w 为“至少一个白格在最大独立集”的行的集合，r_b 为“至少一个黑格在最大独立集”的行集合，c_w 为“至少一个白格在最大独立集”的列的集合，c_b 为“至少一个黑格在最大独立集”的列的集合。
显然的，$r_w \cap r_b = c_w \cap c_b = \varnothing$，因为这东西要是最大独立集。最大独立集的白色状态一定在 $r_w \times c_w$ 中，黑色一定在 $r_b \times c_b$ 中。

无来源题 - Solution

考虑 最大匹配 $=$ 总点数 $-$ 最大独立集，那么删去初始点后最大匹配变化等价于最大独立集变化。

记 r_w 为“至少一个白格在最大独立集”的行的集合，r_b 为“至少一个黑格在最大独立集”的行集合，c_w 为“至少一个白格在最大独立集”的列的集合，c_b 为“至少一个黑格在最大独立集”的列的集合。

显然的，$r_w \cap r_b = c_w \cap c_b = \varnothing$，因为这东西要是最大独立集。最大独立集的白色状态一定在 $r_w \times c_w$ 中，黑色一定在 $r_b \times c_b$ 中。

考虑由于数组有序，那么钦定 r_w 尽量靠前、r_b 尽量靠后一定不劣——因为这样有更多的选择。列同理。

无来源题 - Solution

考虑 最大匹配 $=$ 总点数 $-$ 最大独立集，那么删去初始点后最大匹配变化等价于最大独立集变化。
记 r_w 为“至少一个白格在最大独立集”的行的集合，r_b 为“至少一个黑格在最大独立集”的行集合，c_w 为“至少一个白格在最大独立集”的列的集合，c_b 为“至少一个黑格在最大独立集”的列的集合。
显然的，$r_w \cap r_b = c_w \cap c_b = \varnothing$，因为这东西要是最大独立集。最大独立集的白色状态一定在 $r_w \times c_w$ 中，黑色一定在 $r_b \times c_b$ 中。
考虑由于数组有序，那么钦定 r_w 尽量靠前、r_b 尽量靠后一定不劣——因为这样有更多的选择。列同理。
这个时候终于可以 check 答案了。令 r, c 为行、列的分界线，钦定 $r_w = \{rows | rows \le r\}, r_b = \{rows | rows > r\}$，$c$ 同理。枚举 r，由于各种单调可以均摊 $\mathcal{O}(1)$ 维护 c，动态计算最大独立集也不是很 (zhong) 难 (dian)。
挖掉起点只需要讨论一下起点的位置就可以了。

还看锤子，完了

CraZYali

2020 年 10 月 26 日