博弈论

lk

博弈论

博弈论

- 博弈论，是经济学的一个分支，主要研究具有竞争或对抗性质的对象，在一定规则下产生的各种行为。博弈论考虑游戏中的个体的预测行为和实际行为，并研究它们的优化策略。

博弈论

- 博弈论，是经济学的一个分支，主要研究具有竞争或对抗性质的对象，在一定规则下产生的各种行为。博弈论考虑游戏中的个体的预测行为和实际行为，并研究它们的优化策略。
- 通俗地讲，博弈论主要研究的是：在一个游戏中，进行游戏的多位玩家的策略。

博弈论

- 博弈论，是经济学的一个分支，主要研究具有竞争或对抗性质的对象，在一定规则下产生的各种行为。博弈论考虑游戏中的个体的预测行为和实际行为，并研究它们的优化策略。
- 通俗地讲，博弈论主要研究的是：在一个游戏中，进行游戏的多位玩家的策略。
- (摘自OI-wiki)

博弈

博弈

- OI中博弈通常是指一个游戏，两个玩家轮流进行操作，每个玩家的操作可以使局面变成某些状态中的一个，如果不能进行操作游戏结束。

博弈

- OI中博弈通常是指一个游戏，两个玩家轮流进行操作，每个玩家的操作可以使局面变成某些状态中的一个，如果不能进行操作游戏结束。
- 如果双方都知道完整局面且可以进行的操作相同，称之为平等博弈(否则称之为不平等博弈)

博弈

- OI中博弈通常是指一个游戏，两个玩家轮流进行操作，每个玩家的操作可以使局面变成某些状态中的一个，如果不能进行操作游戏结束。
- 如果双方都知道完整局面且可以进行的操作相同，称之为平等博弈(否则称之为不平等博弈)
- 如果一个局面不会出现两次，显然游戏一定会结束，称之为有限博弈。

博弈

- OI中博弈通常是指一个游戏，两个玩家轮流进行操作，每个玩家的操作可以使局面变成某些状态中的一个，如果不能进行操作游戏结束。
- 如果双方都知道完整局面且可以进行的操作相同，称之为平等博弈(否则称之为不平等博弈)
- 如果一个局面不会出现两次，显然游戏一定会结束，称之为有限博弈。
- 通常认为双方都足够聪明。

有向图游戏

有向图游戏

- 通常的有限平等博弈都可以抽象为在一张dag上，双方可以选择沿某条出边移动棋子。

有向图游戏

- 通常的有限平等博弈都可以抽象为在一张dag上，双方可以选择沿某条出边移动棋子。
- 可以用「必胜态/必败态」表示一个从某一个点开始游戏先手是否必胜(有限博弈中一定先手必胜或后手必胜)

有向图游戏

- 通常的有限平等博弈都可以抽象为在一张dag上，双方可以选择沿某条出边移动棋子。
- 可以用「必胜态/必败态」表示一个从某一个点开始游戏先手是否必胜(有限博弈中一定先手必胜或后手必胜)
- 如果一个点没有出边，根据具体要求判断必胜必败，否则一个点是必胜态当且仅当存在某个出边能到达必败态，必败当且仅当所有出边都是必胜态。

取石子游戏(NIM游戏)

- 有n堆石子，第i堆有a_i个。
- 双方每次可以选择一堆并拿走任意正整数个，无法操作者输。

取石子游戏

- $\bigoplus_{i=1}^{n} a_i = 0$时先手必败，否则先手必胜。
- 证明归纳可得。

SG函数

SG函数

- 现在考虑的是「不能进行操作者负」的规则。

SG函数

- 现在考虑的是「不能进行操作者负」的规则。
- 定义$\mathrm{mex}(S) = \min\{x\}(x \in N, x \notin S)$

SG函数

- 现在考虑的是「不能进行操作者负」的规则。
- 定义$\operatorname{mex}(S) = \min\{x\}(x \in N, x \notin S)$
- 对于每个点x，设它的出边能到达$y_1 \cdots y_k$这些点，则定义$SG(x) = \operatorname{mex} SG(y_1), SG(y_2) \cdots SG(y_k)$。

SG函数

- 现在考虑的是「不能进行操作者负」的规则。
- 定义$\mathrm{mex}(S) = \min\{x\}(x \in N, x \notin S)$
- 对于每个点x,设它的出边能到达$y_1 \cdots y_k$这些点,则定义$SG(x) = \mathrm{mex}\, SG(y_1), SG(y_2) \cdots SG(y_k)$。
- x为起点必胜当且仅当$SG(x) \neq 0$。

SG函数

- 现在考虑的是「不能进行操作者负」的规则。
- 定义$\mathrm{mex}(S) = \min\{x\}(x \in N, x \notin S)$
- 对于每个点x，设它的出边能到达$y_1 \cdots y_k$这些点，则定义$SG(x) = \mathrm{mex}\, SG(y_1), SG(y_2) \cdots SG(y_k)$。
- x为起点必胜当且仅当$SG(x) \neq 0$。
- SG定理：对于多个游戏组合的情况，每次可以选一个图并进行一步操作，则$SG(a_1, a_2 \cdots a_k) = \bigoplus_{i=1}^{n} SG(a_i)$。

PREFIX-FREE GAME

PREFIX-FREE GAME

- 定义一个01串集合是好的，如果每个串长度都在$1 \ldots L$之间，且不存在一个串是另一个串的前缀。

PREFIX-FREE GAME

- 定义一个01串集合是好的，如果每个串长度都在$1 \ldots L$之间，且不存在一个串是另一个串的前缀。
- 有一个「好的」初始01串集合，双方每次需要插入一个新的01串，使得还是个好集合，不能操作者输。

PREFIX-FREE GAME

- 定义一个01串集合是好的，如果每个串长度都在$1 \ldots L$之间，且不存在一个串是另一个串的前缀。
- 有一个「好的」初始01串集合，双方每次需要插入一个新的01串，使得还是个好集合，不能操作者输。
- 求胜者。

PREFIX-FREE GAME

- 定义一个01串集合是好的，如果每个串长度都在$1 \dots L$之间，且不存在一个串是另一个串的前缀。
- 有一个「好的」初始01串集合，双方每次需要插入一个新的01串，使得还是个好集合，不能操作者输。
- 求胜者。
- $1 \le L \le 10^{18}, \sum |s_i| \le 10^5$。

PREFIX-FREE GAME

PREFIX-FREE GAME

- 显然可以把局面拆成一些「初始是空集合，串长小于c，可以选空串」的问题，使得两两独立，现在考虑算一个子问题的SG值。

PREFIX-FREE GAME

- 显然可以把局面拆成一些「初始是空集合，串长小于c，可以选空串」的问题，使得两两独立，现在考虑算一个子问题的SG值。
- $sg(1) = 1, sg(c) = \mathrm{mex}_{i=1}^{c}(\bigoplus_{i}^{c-1} sg(i))$。

PREFIX-FREE GAME

- 显然可以把局面拆成一些「初始是空集合，串长小于c，可以选空串」的问题，使得两两独立，现在考虑算一个子问题的SG值。
- $sg(1) = 1, sg(c) = \text{mex}_{i=1}^{c}(\bigoplus_{i}^{c-1} sg(i))$。
- 打表可得$sg(c) = \text{lowbit}(c)$。

分裂游戏

分裂游戏

- n堆式子，双方可以每次从一堆拿走一个，并往它后面的堆里放入共计两个，不能行动者负。

分裂游戏

- n堆式子，双方可以每次从一堆拿走一个，并往它后面的堆里放入共计两个，不能行动者负。
- 求先手是否能赢，第一步有多少种方案能必胜并求一个字典序最小的第一步方案。

分裂游戏

- n堆式子，双方可以每次从一堆拿走一个，并往它后面的堆里放入共计两个，不能行动者负。
- 求先手是否能赢，第一步有多少种方案能必胜并求一个字典序最小的第一步方案。
- $n \le 21, a_i \le 10000$。

分裂游戏

分裂游戏

- 显然每个石子独立，可以拆开算sg值。

数组游戏

数组游戏

- 有n个格子，m个是白格其它是黑格。

数组游戏

- 有n个格子，m个是白格其它是黑格。
- 每次可以选一个白格x，然后选一个k使得$kx \leq n$，把$x, 2x, \cdots kx$都翻转一遍。

数组游戏

- 有n个格子，m个是白格其它是黑格。
- 每次可以选一个白格x，然后选一个k使得$kx \leq n$，把$x, 2x, \cdots kx$都翻转一遍。
- 无法行动者输。

数组游戏

- 有n个格子,m个是白格其它是黑格。
- 每次可以选一个白格x,然后选一个k使得$kx \leq n$,把$x, 2x, \cdots kx$都翻转一遍。
- 无法行动者输。
- 求先手是否必胜。

数组游戏

- 有n个格子，m个是白格其它是黑格。
- 每次可以选一个白格x，然后选一个k使得$kx \le n$，把$x, 2x, \cdots kx$都翻转一遍。
- 无法行动者输。
- 求先手是否必胜。
- $n \le 10^9, m \le 100$。

数组游戏

数组游戏

- 显然每个白格独立，可以拆开算。

数组游戏

- 显然每个白格独立，可以拆开算。
- 直接dp的复杂度是$O(n\log n)$的，跑不过。

数组游戏

- 显然每个白格独立，可以拆开算。
- 直接dp的复杂度是$O(n\log n)$的，跑不过。
- 注意到sg[x]只和$\lfloor \frac{n}{x} \rfloor$相关，可以不重复算。

数组游戏

- 显然每个白格独立，可以拆开算。
- 直接dp的复杂度是$O(n\log n)$的，跑不过。
- 注意到sg[x]只和$\lfloor \frac{n}{x} \rfloor$相关，可以不重复算。
- 直接整除分块套整除分块就能过了。

GAMES

GAMES

- 有n堆石子，第i堆有a_i个。

GAMES

- 有n堆石子，第i堆有a_i个。
- 双方每次可以选择不超过6堆并拿走任意正整数个，无法操作者输。

GAMES

- 有n堆石子，第i堆有a_i个。
- 双方每次可以选择不超过6堆并拿走任意正整数个，无法操作者输。
- 现在每一个a_i都可以从$b_1 \dots b_m$里选，求有多少种方案先手必败。

GAMES

- 有n堆石子,第i堆有a_i个。
- 双方每次可以选择不超过6堆并拿走任意正整数个,无法操作者输。
- 现在每一个a_i都可以从$b_1 \ldots b_m$里选,求有多少种方案先手必败。
- $\bmod 998244353, b_i \leq 100$

GAMES

GAMES

- 考虑怎样先手必败。

GAMES

- 考虑怎样先手必败。
- 这个问题和一般的Nim游戏的区别是最多选6堆而不是1堆，1堆的情况下必败条件是异或和=0。

GAMES

- 考虑怎样先手必败。
- 这个问题和一般的Nim游戏的区别是最多选6堆而不是1堆，1堆的情况下必败条件是异或和=0。
- 异或和$= 0$也就是每个二进制位出现了偶数次，大胆猜想6堆的情况必败条件是每一位出现了7倍数次。

GAMES

- 考虑怎样先手必败。
- 这个问题和一般的Nim游戏的区别是最多选6堆而不是1堆, 1堆的情况下必败条件是异或和=0。
- 异或和$=0$也就是每个二进制位出现了偶数次, 大胆猜想6堆的情况必败条件是每一位出现了7倍数次。
- 归纳可证。

GAMES

- 考虑怎样先手必败。
- 这个问题和一般的Nim游戏的区别是最多选6堆而不是1堆，1堆的情况下必败条件是异或和=0。
- 异或和$=0$也就是每个二进制位出现了偶数次，大胆猜想6堆的情况必败条件是每一位出现了7倍数次。
- 归纳可证。
- 计数的情况，$998244353 = 7 \times 17 \times 2^{23} + 1$，所以存在7次单位根。而$\lg 100 = 7, 7^7 = 823543$。

GAMES

- 考虑怎样先手必败。
- 这个问题和一般的Nim游戏的区别是最多选6堆而不是1堆，1堆的情况下必败条件是异或和=0。
- 异或和$=0$也就是每个二进制位出现了偶数次，大胆猜想6堆的情况必败条件是每一位出现了7倍数次。
- 归纳可证。
- 计数的情况，$998244353 = 7 \times 17 \times 2^{23} + 1$，所以存在7次单位根。而$\lg 100 = 7, 7^7 = 823543$。
- 所以写个7进制FWT就可以了。

移动金币

移动金币

- 有个$1 \times n$的棋盘上有m枚金币，位置两两不同。

移动金币

- 有个$1 \times n$的棋盘上有m枚金币,位置两两不同。
- 双方每次可以选一枚金币往左移动任意多(> 0)格,不能移出棋盘或越过其他金币(也不能和别的金币位置一样)。

移动金币

- 有个$1 \times n$的棋盘上有m枚金币,位置两两不同。
- 双方每次可以选一枚金币往左移动任意多(> 0)格,不能移出棋盘或越过其他金币(也不能和别的金币位置一样)。
- 无法操作者输。

移动金币

- 有个$1 \times n$的棋盘上有m枚金币，位置两两不同。
- 双方每次可以选一枚金币往左移动任意多(> 0)格，不能移出棋盘或越过其他金币(也不能和别的金币位置一样)。
- 无法操作者输。
- 求先手必胜的初始局面数。

移动金币

- 有个$1 \times n$的棋盘上有m枚金币，位置两两不同。
- 双方每次可以选一枚金币往左移动任意多(> 0)格，不能移出棋盘或越过其他金币(也不能和别的金币位置一样)。
- 无法操作者输。
- 求先手必胜的初始局面数。
- $n \leq 150000, m \leq 50$

移动金币

- 有个$1 \times n$的棋盘上有m枚金币,位置两两不同。
- 双方每次可以选一枚金币往左移动任意多(> 0)格,不能移出棋盘或越过其他金币(也不能和别的金币位置一样)。
- 无法操作者输。
- 求先手必胜的初始局面数。
- $n \leq 150000, m \leq 50$
- 加强:$n, m \leq 1000000$

移动金币

移动金币

- 把$1 \times n$的棋盘m枚金币转换为$n - m$个石子分成$m + 1$堆, 下标从$0 \ldots m$。

移动金币

- 把$1 \times n$的棋盘m枚金币转换为$n - m$个石子分成$m + 1$堆, 下标从$0 \dots m$。
- 那么每次可以把一堆移到上一堆, 全移到第0堆时结束。

移动金币

- 把$1 \times n$的棋盘m枚金币转换为$n - m$个石子分成$m + 1$堆, 下标从$0 \ldots m$。
- 那么每次可以把一堆移到上一堆, 全移到第0堆时结束。
- 那么可以猜到, 先手必败当且仅当奇数堆的xor和是0, 并且可以简单归纳证明。

移动金币

- 把$1 \times n$的棋盘m枚金币转换为$n - m$个石子分成$m + 1$堆，下标从$0 \ldots m$。
- 那么每次可以把一堆移到上一堆，全移到第0堆时结束。
- 那么可以猜到，先手必败当且仅当奇数堆的xor和是0，并且可以简单归纳证明。
- 把先手必胜改成求先手必败的方案数，那么问题就是求奇数堆xor和是0的方案数。

移动金币

移动金币

- 先不管偶数堆，算出每个奇数堆sum的方案数，乘个组合数就行。

移动金币

- 先不管偶数堆，算出每个奇数堆sum的方案数，乘个组合数就行。
- 从高位到低位dp，每次往最后加一位，然后枚举有多少堆这个位是1。

移动金币

- 先不管偶数堆，算出每个奇数堆sum的方案数，乘个组合数就行。
- 从高位到低位dp，每次往最后加一位，然后枚举有多少堆这个位是1。
- 假设有k个奇数($k = \lceil \frac{m}{2} \rceil$)堆。

移动金币

- 先不管偶数堆，算出每个奇数堆sum的方案数，乘个组合数就行。
- 从高位到低位dp，每次往最后加一位，然后枚举有多少堆这个位是1。
- 假设有k个奇数($k = \lceil \frac{m}{2} \rceil$)堆。
- 转成生成函数就是$F \leftarrow F(x^2)G$，其中$G = \sum_{2|i, 0 \le i \le k} x^i \binom{k}{i}$。

移动金币

- 先不管偶数堆，算出每个奇数堆sum的方案数，乘个组合数就行。
- 从高位到低位dp，每次往最后加一位，然后枚举有多少堆这个位是1。
- 假设有k个奇数($k = \lceil \frac{m}{2} \rceil$)堆。
- 转成生成函数就是$F \leftarrow F(x^2)G$，其中$G = \sum_{2|i,0\le i\le k} x^i \binom{k}{i}$。
- 这样做的复杂度是$O(n\log^2 n)$。

移动金币

- 先不管偶数堆，算出每个奇数堆sum的方案数，乘个组合数就行。
- 从高位到低位dp，每次往最后加一位，然后枚举有多少堆这个位是1。
- 假设有k个奇数($k=\lceil\frac{m}{2}\rceil$)堆。
- 转成生成函数就是$F\leftarrow F(x^2)G$，其中$G=\sum_{2|i,0\leq i\leq k}x^i\binom{k}{i}$。
- 这样做的复杂度是$O(n\log^2 n)$。
- 显然最后一次$F\leftarrow F(x^2)G$之后要用的位数是$n-m$，那么之前要用的位数就是$\lfloor\frac{n-m}{2}\rfloor$。

移动金币

- 先不管偶数堆，算出每个奇数堆sum的方案数，乘个组合数就行。
- 从高位到低位dp，每次往最后加一位，然后枚举有多少堆这个位是1。
- 假设有k个奇数($k = \lceil \frac{m}{2} \rceil$)堆。
- 转成生成函数就是$F \leftarrow F(x^2)G$，其中$G = \sum_{2|i, 0 \le i \le k} x^i \binom{k}{i}$。
- 这样做的复杂度是$O(n \log^2 n)$。
- 显然最后一次$F \leftarrow F(x^2)G$之后要用的位数是$n - m$，那么之前要用的位数就是$\lfloor \frac{n-m}{2} \rfloor$。
- 以此类推，每往前一次需要的位数都会$/2$，那么如果只记有效长度，每次fft的时候长度都会翻倍。

移动金币

- 先不管偶数堆，算出每个奇数堆sum的方案数，乘个组合数就行。
- 从高位到低位dp，每次往最后加一位，然后枚举有多少堆这个位是1。
- 假设有k个奇数($k = \lceil \frac{m}{2} \rceil$)堆。
- 转成生成函数就是$F \leftarrow F(x^2)G$，其中$G = \sum_{2|i,0\leq i\leq k} x^i \binom{k}{i}$。
- 这样做的复杂度是$O(n\log^2 n)$。
- 显然最后一次$F \leftarrow F(x^2)G$之后要用的位数是$n-m$，那么之前要用的位数就是$\lfloor \frac{n-m}{2} \rfloor$。
- 以此类推，每往前一次需要的位数都会$/2$，那么如果只记有效长度，每次fft的时候长度都会翻倍。
- 时间复杂度$O(n\log n)$。

HUNGERGAME

HUNGERGAME

- n个盒子，第i个盒子里有a_i个石子(双方最初已知a_i)，且全部没打开。

HUNGERGAME

- n个盒子,第i个盒子里有a_i个石子(双方最初已知a_i),且全部没打开。
- 每次可以拆开一些盒子或者从某一个打开的盒子中的石子取出一些。

HUNGERGAME

- n个盒子，第i个盒子里有a_i个石子(双方最初已知a_i)，且全部没打开。
- 每次可以拆开一些盒子或者从某一个打开的盒子中的石子取出一些。
- 求先手是否必胜。

HUNGERGAME

- n个盒子，第i个盒子里有a_i个石子(双方最初已知a_i)，且全部没打开。
- 每次可以拆开一些盒子或者从某一个打开的盒子中的石子取出一些。
- 求先手是否必胜。
- $n \le 20$。

HUNGERGAME

HUNGERGAME

- 考虑一个状态啥时候能赢。

HUNGERGAME

- 考虑一个状态啥时候能赢。
- 首先有一个平凡的情况：打开的盒子xor和为0且剩下的盒子xor和为0，此时先手只要全打开一定能赢。

HUNGERGAME

- 考虑一个状态啥时候能赢。
- 首先有一个平凡的情况:打开的盒子xor和为0且剩下的盒子xor和为0,此时先手只要全打开一定能赢。
- 再接着考虑打开的盒子xor和为0的情况。如果存在一个子集xor和为0,那么直接把这个子集全打开,并且不断这么操作直到不存在子集,会给对面一个不存在xor和为0的情况。

HUNGERGAME

- 考虑一个状态啥时候能赢。
- 首先有一个平凡的情况:打开的盒子xor和为0且剩下的盒子xor和为0,此时先手只要全打开一定能赢。
- 再接着考虑打开的盒子xor和为0的情况。如果存在一个子集xor和为0,那么直接把这个子集全打开,并且不断这么操作直到不存在子集,会给对面一个不存在xor和为0的情况。
- 对于不存在没打开盒子的子集xor和为0的情况,不管你怎么操作对面都可以在一步操作内变回来,所以必败。

HUNGERGAME

- 考虑一个状态啥时候能赢。
- 首先有一个平凡的情况:打开的盒子xor和为0且剩下的盒子xor和为0,此时先手只要全打开一定能赢。
- 再接着考虑打开的盒子xor和为0的情况。如果存在一个子集xor和为0,那么直接把这个子集全打开,并且不断这么操作直到不存在子集,会给对面一个不存在xor和为0的情况。
- 对于不存在没打开盒子的子集xor和为0的情况,不管你怎么操作对面都可以在一步操作内变回来,所以必败。
- ~~什么,打开的盒子xor和不为0的情况?和这题有关系吗?~~

HUNGERGAME

- 考虑一个状态啥时候能赢。
- 首先有一个平凡的情况：打开的盒子xor和为0且剩下的盒子xor和为0，此时先手只要全打开一定能赢。
- 再接着考虑打开的盒子xor和为0的情况。如果存在一个子集xor和为0，那么直接把这个子集全打开，并且不断这么操作直到不存在子集，会给对面一个不存在xor和为0的情况。
- 对于不存在没打开盒子的子集xor和为0的情况，不管你怎么操作对面都可以在一步操作内变回来，所以必败。
- ~~什么，打开的盒子xor和不为0的情况？和这题有关系吗？~~
- 直接判判有没有子集xor和为0就行。

HUNGERGAME

- 考虑一个状态啥时候能赢。
- 首先有一个平凡的情况：打开的盒子xor和为0且剩下的盒子xor和为0，此时先手只要全打开一定能赢。
- 再接着考虑打开的盒子xor和为0的情况。如果存在一个子集xor和为0，那么直接把这个子集全打开，并且不断这么操作直到不存在子集，会给对面一个不存在xor和为0的情况。
- 对于不存在没打开盒子的子集xor和为0的情况，不管你怎么操作对面都可以在一步操作内变回来，所以必败。
- ~~什么，打开的盒子xor和不为0的情况？和这题有关系吗？~~
- 直接判判有没有子集xor和为0就行。
- 时间复杂度$O(n\log a_i)$。

ANTI-NIM

ANTI-NIM

- 有n堆石子，第i堆有a_i个。

ANTI-NIM

- 有n堆石子，第i堆有a_i个。
- 双方每次可以选择一堆并拿走任意正整数个，无法操作者胜。

ANTI-NIM

ANTI-NIM

- 如果每一堆个数都是1,则先手必胜当且仅当有偶数堆(SG值为0)

ANTI-NIM

- 如果每一堆个数都是1,则先手必胜当且仅当有偶数堆(SG值为0)
- 否则SG值$\neq 0$的情况下:

ANTI-NIM

- 如果每一堆个数都是1,则先手必胜当且仅当有偶数堆(SG值为0)
- 否则SG值$\neq 0$的情况下:
- 如果有恰好一堆≥ 2,则先手可以取这堆变成0或1,使得剩下奇数堆1,此时必胜。

ANTI-NIM

- 如果每一堆个数都是1, 则先手必胜当且仅当有偶数堆(SG值为0)
- 否则SG值$\neq 0$的情况下:
- 如果有恰好一堆≥ 2, 则先手可以取这堆变成0或1, 使得剩下奇数堆1, 此时必胜。
- 否则先手一定可以将它变为$SG = 0$, 且此时至少有一堆≥ 2。

ANTI-NIM

- 如果每一堆个数都是1,则先手必胜当且仅当有偶数堆(SG值为0)
- 否则SG值$\neq 0$的情况下:
- 如果有恰好一堆≥ 2,则先手可以取这堆变成0或1,使得剩下奇数堆1,此时必胜。
- 否则先手一定可以将它变为$SG = 0$,且此时至少有一堆≥ 2。
- 如果SG值$= 0$,则至少有两堆≥ 2,任意操作一定会变成至少一堆≥ 2且$SG \neq 0$。

ANTI-NIM

- 如果每一堆个数都是1,则先手必胜当且仅当有偶数堆(SG值为0)
- 否则SG值$\neq 0$的情况下:
- 如果有恰好一堆≥ 2,则先手可以取这堆变成0或1,使得剩下奇数堆1,此时必胜。
- 否则先手一定可以将它变为$SG = 0$,且此时至少有一堆≥ 2。
- 如果SG值$= 0$,则至少有两堆≥ 2,任意操作一定会变成至少一堆≥ 2且$SG \neq 0$。
- 所以先手必胜当且仅当:

ANTI-NIM

- 如果每一堆个数都是1,则先手必胜当且仅当有偶数堆(SG值为0)
- 否则SG值$\neq 0$的情况下:
- 如果有恰好一堆≥ 2,则先手可以取这堆变成0或1,使得剩下奇数堆1,此时必胜。
- 否则先手一定可以将它变为$SG = 0$,且此时至少有一堆≥ 2。
- 如果SG值$= 0$,则至少有两堆≥ 2,任意操作一定会变成至少一堆≥ 2且$SG \neq 0$。
- 所以先手必胜当且仅当:

$$1. SG = 0\text{且每一堆个数都是}1\text{。}$$

ANTI-NIM

- 如果每一堆个数都是1,则先手必胜当且仅当有偶数堆(SG值为0)
- 否则SG值$\neq 0$的情况下:
- 如果有恰好一堆≥ 2,则先手可以取这堆变成0或1,使得剩下奇数堆1,此时必胜。
- 否则先手一定可以将它变为$SG = 0$,且此时至少有一堆≥ 2。
- 如果SG值$= 0$,则至少有两堆≥ 2,任意操作一定会变成至少一堆≥ 2且$SG \neq 0$。
- 所以先手必胜当且仅当:

1. $SG = 0$且每一堆个数都是1。
2. $SG \neq 0$且至少有一堆≥ 1。

TREE-TAC-TOE

TREE-TAC-TOE

- 给定一个树，有些点被染成了白色，其它点没有染色。

TREE-TAC-TOE

- 给定一个树，有些点被染成了白色，其它点没有染色。
- Alice执白，Bob执黑，Alice先手。双方每次可以把一个无色点染成自己颜色，如果出现一条长度为3的同色路径则胜利，如果到最后还没有则平局。

TREE-TAC-TOE

- 给定一个树，有些点被染成了白色，其它点没有染色。
- Alice执白，Bob执黑，Alice先手。双方每次可以把一个无色点染成自己颜色，如果出现一条长度为3的同色路径则胜利，如果到最后还没有则平局。
- 保证初始局面白点还没有直接胜利。

TREE-TAC-TOE

- 给定一个树，有些点被染成了白色，其它点没有染色。
- Alice执白，Bob执黑，Alice先手。双方每次可以把一个无色点染成自己颜色，如果出现一条长度为3的同色路径则胜利，如果到最后还没有则平局。
- 保证初始局面白点还没有直接胜利。
- 求胜负情况。

TREE-TAC-TOE

- 给定一个树，有些点被染成了白色，其它点没有染色。
- Alice执白，Bob执黑，Alice先手。双方每次可以把一个无色点染成自己颜色，如果出现一条长度为3的同色路径则胜利，如果到最后还没有则平局。
- 保证初始局面白点还没有直接胜利。
- 求胜负情况。
- $n \leq 5 \times 10^5$。

TREE-TAC-TOE

TREE-TAC-TOE

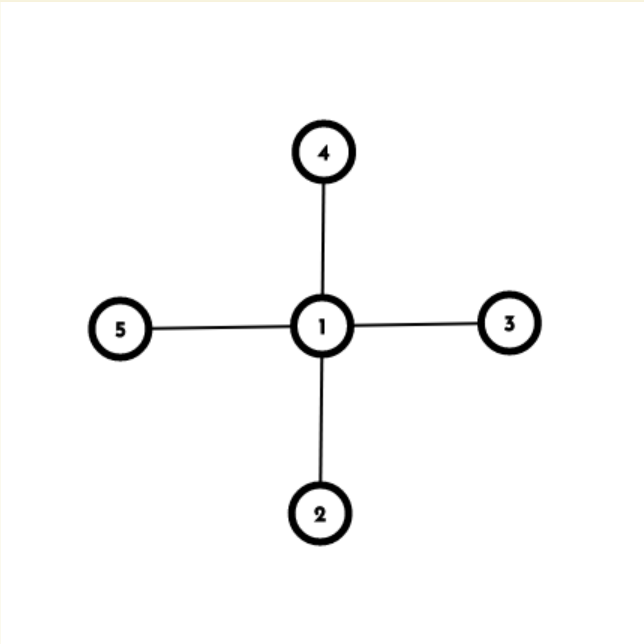
- 先手一定不败。

TREE-TAC-TOE

- 先手一定不败。
- 考虑没有白色点的情况，如果出现以下三种子图则先手可以通过放在1号点位置的点直接胜利：

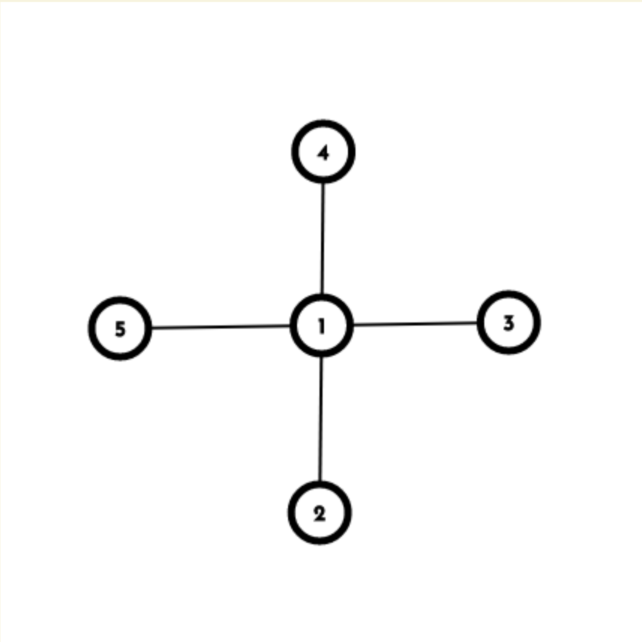
TREE-TAC-TOE

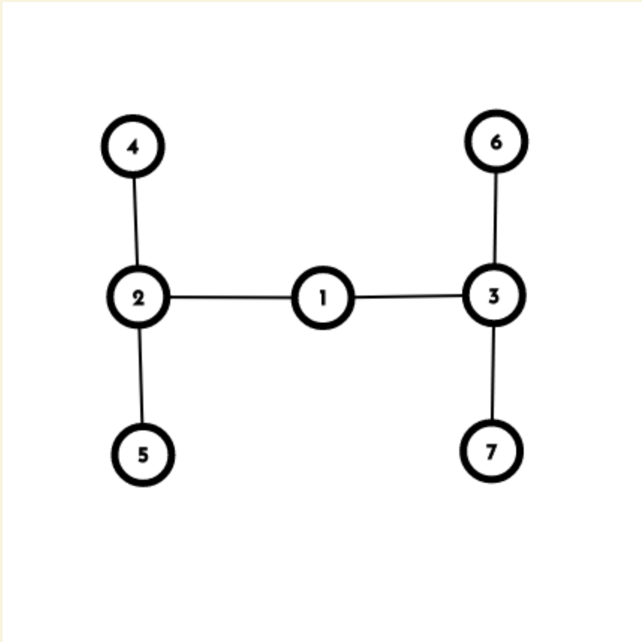
- 先手一定不败。
- 考虑没有白色点的情况，如果出现以下三种子图则先手可以通过放在1号点位置的点直接胜利：



TREE-TAC-TOE

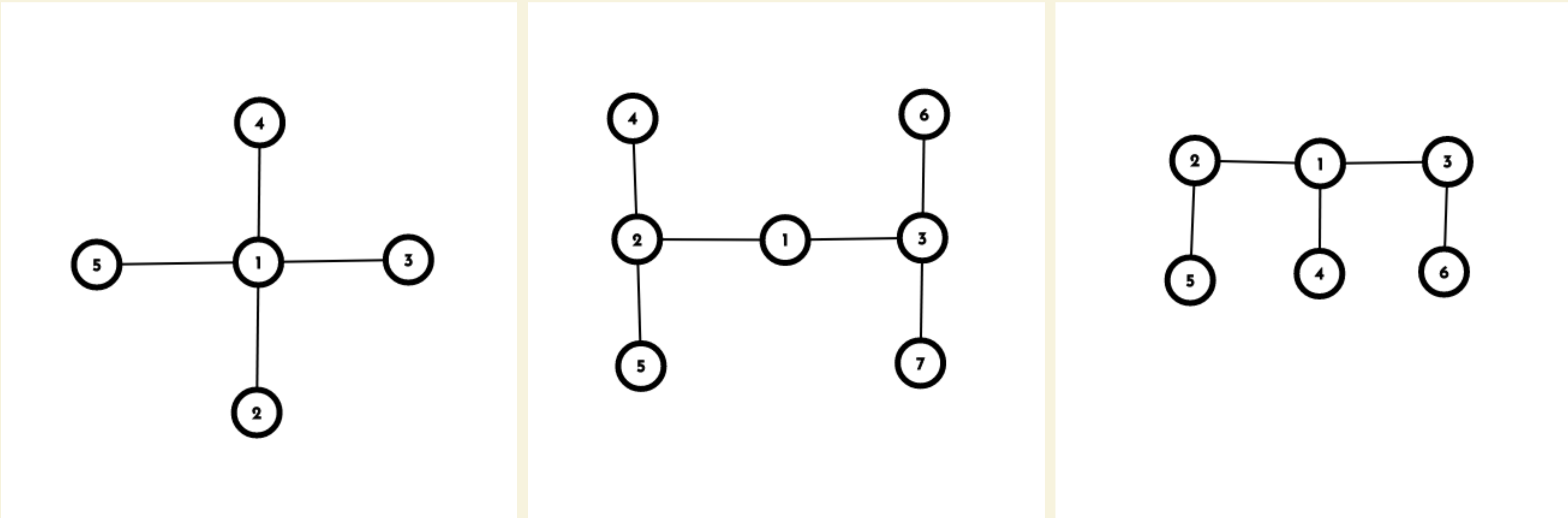
- 先手一定不败。
- 考虑没有白色点的情况，如果出现以下三种子图则先手可以通过放在1号点位置的点直接胜利：





TREE-TAC-TOE

- 先手一定不败。
- 考虑没有白色点的情况，如果出现以下三种子图则先手可以通过放在1号点位置的点直接胜利：



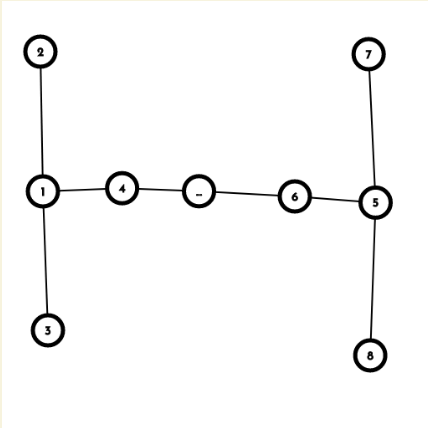
TREE-TAC-TOE

TREE-TAC-TOE

- 对于没有这几种子图的情况，图大概长成这样：

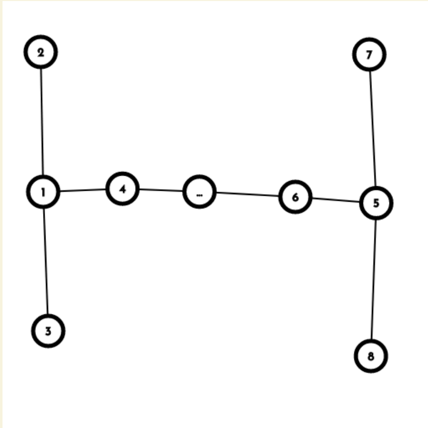
TREE-TAC-TOE

- 对于没有这几种子图的情况，图大概长成这样：



TREE-TAC-TOE

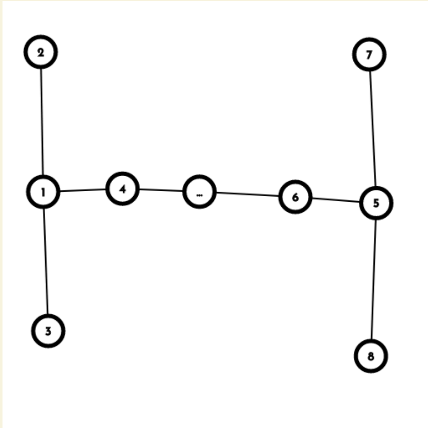
- 对于没有这几种子图的情况，图大概长成这样：



- 中间是一个长度≥ 0的链。当然也可能少掉7,8退化成一个Y字形，下面长度≥ 1就行。也可能少掉1,2,7,8退化成长度≥ 1的链，也有可能中间退化了变成一个H。

TREE-TAC-TOE

- 对于没有这几种子图的情况，图大概长成这样：



- 中间是一个长度≥ 0的链。当然也可能少掉7,8退化成一个Y字形，下面长度≥ 1就行。也可能少掉1,2,7,8退化成长度≥ 1的链，也有可能中间退化了变成一个H。
- 后三种情况先手显然赢不了了，第一种也一样。。。吗？

TREE-TAC-TOE

TREE-TAC-TOE

- 考虑存在白色点的情况，显然对白方不会劣于全无色，那么只要考虑没有那三个子图的情况能不能变成赢。

TREE-TAC-TOE

- 考虑存在白色点的情况，显然对白方不会劣于全无色，那么只要考虑没有那三个子图的情况能不能变成赢。
- 先判掉放一个点就直接获胜的情况。

TREE-TAC-TOE

- 考虑存在白色点的情况，显然对白方不会劣于全无色，那么只要考虑没有那三个子图的情况能不能变成赢。
- 先判掉放一个点就直接获胜的情况。
- 如果一个初始白点是三叉点或者邻点是三叉点或者有两个邻点且至少一个有其它出边，显然白必胜了。

TREE-TAC-TOE

- 考虑存在白色点的情况，显然对白方不会劣于全无色，那么只要考虑没有那三个子图的情况能不能变成赢。
- 先判掉放一个点就直接获胜的情况。
- 如果一个初始白点是三叉点或者邻点是三叉点或者有两个邻点且至少一个有其它出边，显然白必胜了。
- 如果所有白点都不是，但是有白点还有几种情况：

TREE-TAC-TOE

- 考虑存在白色点的情况，显然对白方不会劣于全无色，那么只要考虑没有那三个子图的情况能不能变成赢。
- 先判掉放一个点就直接获胜的情况。
- 如果一个初始白点是三叉点或者邻点是三叉点或者有两个邻点且至少一个有其它出边，显然白必胜了。
- 如果所有白点都不是，但是有白点还有几种情况：
- 一个链，一端是白，显然白赢不了。一个链，两端是白，看上去不行?

TREE-TAC-TOE

- 考虑存在白色点的情况，显然对白方不会劣于全无色，那么只要考虑没有那三个子图的情况能不能变成赢。
- 先判掉放一个点就直接获胜的情况。
- 如果一个初始白点是三叉点或者邻点是三叉点或者有两个邻点且至少一个有其它出边，显然白必胜了。
- 如果所有白点都不是，但是有白点还有几种情况：
- 一个链，一端是白，显然白赢不了。一个链，两端是白，看上去不行？
- 实际上比如一个长度为5两端白的链，选个中间就保证胜利了，，，

TREE-TAC-TOE

- 考虑存在白色点的情况，显然对白方不会劣于全无色，那么只要考虑没有那三个子图的情况能不能变成赢。
- 先判掉放一个点就直接获胜的情况。
- 如果一个初始白点是三叉点或者邻点是三叉点或者有两个邻点且至少一个有其它出边，显然白必胜了。
- 如果所有白点都不是，但是有白点还有几种情况：
- 一个链，一端是白，显然白赢不了。一个链，两端是白，看上去不行?
- 实际上比如一个长度为5两端白的链，选个中间就保证胜利了，，，
- 推广一下，任何>3的奇数长度两端白的链1..n,每次白选n-2可以强制黑选n-1变成n-2的问题，所以白必胜。

TREE-TAC-TOE

TREE-TAC-TOE

- 对于偶数的情况，长度为2的时候显然先手赢不了。

TREE-TAC-TOE

- 对于偶数的情况，长度为2的时候显然先手赢不了。
- 类似的，对于>=4链，每次白选一个之后都会把链分成一个奇数段和一个偶数段，黑一定能把选的点奇数段那侧那个点选了，这样变成个更小的偶数问题，所以白赢不了。

TREE-TAC-TOE

- 对于偶数的情况，长度为2的时候显然先手赢不了。
- 类似的，对于>=4链，每次白选一个之后都会把链分成一个奇数段和一个偶数段，黑一定能把选的点奇数段那侧那个点选了，这样变成个更小的偶数问题，所以白赢不了。
- 然后Y底下有个点的情况也差不多，每次可以把底下的链长度-2，也就是总点数是偶数的时候白一定能缩成一个「一个三度点有一边是白点共4个点」的图。

TREE-TAC-TOE

- 对于偶数的情况，长度为2的时候显然先手赢不了。
- 类似的，对于>=4链，每次白选一个之后都会把链分成一个奇数段和一个偶数段，黑一定能把选的点奇数段那侧那个点选了，这样变成个更小的偶数问题，所以白赢不了。
- 然后Y底下有个点的情况也差不多，每次可以把底下的链长度-2，也就是总点数是偶数的时候白一定能缩成一个「一个三度点有一边是白点共4个点」的图。
- 那么可以用类似的方式解决前面的问题：对于一个H字形，只有这6个点的情况显然白赢不了了，7个就能赢(认为中间链长度为1?)

TREE-TAC-TOE

- 对于偶数的情况，长度为2的时候显然先手赢不了。
- 类似的，对于>=4链，每次白选一个之后都会把链分成一个奇数段和一个偶数段，黑一定能把选的点奇数段那侧那个点选了，这样变成个更小的偶数问题，所以白赢不了。
- 然后Y底下有个点的情况也差不多，每次可以把底下的链长度-2，也就是总点数是偶数的时候白一定能缩成一个「一个三度点有一边是白点共4个点」的图。
- 那么可以用类似的方式解决前面的问题：对于一个H字形，只有这6个点的情况显然白赢不了了，7个就能赢(认为中间链长度为1?)
- 对于中间链长度是>1的奇数的情况，设中间链是$1\cdots n$，可以先选1让黑色强制选0(也就是前面一个点)，然后按照3,5,7的顺序选，这样黑点只能选2,4,6，直到白点选n的时候黑点就gg了。

TREE-TAC-TOE

TREE-TAC-TOE

- 时间复杂度$O(n)$。

有向グラフと数

有向グラフと数

- 有一个有向图，每个点有个分数。

有向グラフと数

- 有一个有向图，每个点有个分数。
- 现在1号点有个棋子，Alice Bob轮流进行如下操作：移动到某条出边连向的点，或者宣告结束。Alice先。

有向グラフと数

- 有一个有向图，每个点有个分数。
- 现在1号点有个棋子，Alice Bob轮流进行如下操作：移动到某条出边连向的点，或者宣告结束。Alice先。
- 当双方移动10^9步后，若没人宣告结束则自动结束。

有向グラフと数

- 有一个有向图，每个点有个分数。
- 现在1号点有个棋子，Alice Bob轮流进行如下操作：移动到某条出边连向的点，或者宣告结束。Alice先。
- 当双方移动10^9步后，若没人宣告结束则自动结束。
- Alice希望最大化棋子最后位置的分数，Bob希望最小化。

有向グラフと数

- 有一个有向图，每个点有个分数。
- 现在1号点有个棋子，Alice Bob轮流进行如下操作：移动到某条出边连向的点，或者宣告结束。Alice先。
- 当双方移动10^9步后，若没人宣告结束则自动结束。
- Alice希望最大化棋子最后位置的分数，Bob希望最小化。
- 求双方无限聪明的情况下，最后位置的分数。

有向グラフと数

- 有一个有向图，每个点有个分数。
- 现在1号点有个棋子，Alice Bob轮流进行如下操作：移动到某条出边连向的点，或者宣告结束。Alice先。
- 当双方移动10^9步后，若没人宣告结束则自动结束。
- Alice希望最大化棋子最后位置的分数，Bob希望最小化。
- 求双方无限聪明的情况下，最后位置的分数。
- $n \le 10^5, m \le 2 \cdot 10^5$。

有向グラフと数

有向グラフと数

- 首先我们知道答案一定$\geq a_1$,否则Alice开局直接结束即可。

有向グラフと数

- 首先我们知道答案一定$\geq a_1$, 否则Alice开局直接结束即可。
- 二分答案, 把所有点转成01, 最初在0点, 求能否在1点结束。

有向グラフと数

- 首先我们知道答案一定$\geq a_1$，否则Alice开局直接结束即可。
- 二分答案，把所有点转成01，最初在0点，求能否在1点结束。
- 那么Alice一定会从0走到1，Bob会从1走到0，没法走就gg了。

有向グラフと数

- 首先我们知道答案一定$\geq a_1$，否则Alice开局直接结束即可。
- 二分答案，把所有点转成01，最初在0点，求能否在1点结束。
- 那么Alice一定会从0走到1，Bob会从1走到0，没法走就gg了。
- dp[x]表示现在在x，现在该走的人能不能赢。

有向グラフと数

- 首先我们知道答案一定$\geq a_1$，否则Alice开局直接结束即可。
- 二分答案，把所有点转成01，最初在0点，求能否在1点结束。
- 那么Alice一定会从0走到1，Bob会从1走到0，没法走就gg了。
- dp[x]表示现在在x，现在该走的人能不能赢。
- 显然一个点能赢当且仅当存在一个不同颜色(指01)的出边会输，如果所有出边都一定必胜那就得输了。

有向グラフと数

- 首先我们知道答案一定$\geq a_1$,否则Alice开局直接结束即可。
- 二分答案,把所有点转成01,最初在0点,求能否在1点结束。
- 那么Alice一定会从0走到1, Bob会从1走到0,没法走就gg了。
- dp[x]表示现在在x,现在该走的人能不能赢。
- 显然一个点能赢当且仅当存在一个不同颜色(指01)的出边会输,如果所有出边都一定必胜那就得输了。
- 对反图拓扑排序dp就行,注意一个点如果确定是必胜或者必败了就把它加进队列或者栈,不需要把所有入边都判完。

有向グラフと数

- 首先我们知道答案一定$\geq a_1$,否则Alice开局直接结束即可。
- 二分答案,把所有点转成01,最初在0点,求能否在1点结束。
- 那么Alice一定会从0走到1,Bob会从1走到0,没法走就gg了。
- dp[x]表示现在在x,现在该走的人能不能赢。
- 显然一个点能赢当且仅当存在一个不同颜色(指01)的出边会输,如果所有出边都一定必胜那就得输了。
- 对反图拓扑排序dp就行,注意一个点如果确定是必胜或者必败了就把它加进队列或者栈,不需要把所有入边都判完。
- 如果一个点最后没确定状态,那么最优情况下你一步对方一步一直不会停,但你先走,游戏结束的时候是对方走完一步之后,所以你就输了。

有向グラフと数

- 首先我们知道答案一定$\geq a_1$,否则Alice开局直接结束即可。
- 二分答案,把所有点转成01,最初在0点,求能否在1点结束。
- 那么Alice一定会从0走到1, Bob会从1走到0,没法走就gg了。
- dp[x]表示现在在x,现在该走的人能不能赢。
- 显然一个点能赢当且仅当存在一个不同颜色(指01)的出边会输,如果所有出边都一定必胜那就得输了。
- 对反图拓扑排序dp就行,注意一个点如果确定是必胜或者必败了就把它加进队列或者栈,不需要把所有入边都判完。
- 如果一个点最后没确定状态,那么最优情况下你一步对方一步一直不会停,但你先走,游戏结束的时候是对方走完一步之后,所以你就输了。
- 时间复杂度$O((n+m)\log n)$。

GAME ON GRAPH

GAME ON GRAPH

- 有一个有向图，某一个点有个棋子。

GAME ON GRAPH

- 有一个有向图，某一个点有个棋子。
- Alice和Bob轮流进行操作，每次可以将棋子移动到相邻点，不能操作者输。

GAME ON GRAPH

- 有一个有向图，某一个点有个棋子。
- Alice和Bob轮流进行操作，每次可以将棋子移动到相邻点，不能操作者输。
- Alice想游戏无限进行，除此之外更想赢。

GAME ON GRAPH

- 有一个有向图，某一个点有个棋子。
- Alice和Bob轮流进行操作，每次可以将棋子移动到相邻点，不能操作者输。
- Alice想游戏无限进行，除此之外更想赢。
- Bob想赢，除此之外更想输而不是无限进行。

GAME ON GRAPH

- 有一个有向图，某一个点有个棋子。
- Alice和Bob轮流进行操作，每次可以将棋子移动到相邻点，不能操作者输。
- Alice想游戏无限进行，除此之外更想赢。
- Bob想赢，除此之外更想输而不是无限进行。
- 双方无限聪明，求初始棋子在每个点，Alice和Bob先手时 先手赢还是输还是无限进行。

GAME ON GRAPH

- 有一个有向图，某一个点有个棋子。
- Alice和Bob轮流进行操作，每次可以将棋子移动到相邻点，不能操作者输。
- Alice想游戏无限进行，除此之外更想赢。
- Bob想赢，除此之外更想输而不是无限进行。
- 双方无限聪明，求初始棋子在每个点，Alice和Bob先手时 先手赢还是输还是无限进行。
- $n \le 10^5, m \le 2 \cdot 10^5$。

GAME ON GRAPH

GAME ON GRAPH

- 类似的在反图拓扑序dp一下。

GAME ON GRAPH

- 类似的在反图拓扑序dp一下。
- 由于双方策略不同，所以需要dp两个人分别先手的情况。

GAME ON GRAPH

- 类似的在反图拓扑序dp一下。
- 由于双方策略不同，所以需要dp两个人分别先手的情况。
- 如果没有确定胜败就是平局。

GAME ON GRAPH

- 类似的在反图拓扑序dp一下。
- 由于双方策略不同，所以需要dp两个人分别先手的情况。
- 如果没有确定胜败就是平局。
- 需要处理的是，如果第二个人能走到一个必败态就确定是必胜态了，而第一个人所有出边都确定胜败了才知道必胜还是必败。

GAME ON GRAPH

- 类似的在反图拓扑序dp一下。
- 由于双方策略不同，所以需要dp两个人分别先手的情况。
- 如果没有确定胜败就是平局。
- 需要处理的是，如果第二个人能走到一个必败态就确定是必胜态了，而第一个人所有出边都确定胜败了才知道必胜还是必败。
- 当然前面的部分做完了之后，如果第二个人某一个点，走不到必败态但能走到一个必胜态，在他的最优策略下他会直接走过去，这里需要进行另一个拓扑序dp。

GAME ON GRAPH

- 类似的在反图拓扑序dp一下。
- 由于双方策略不同，所以需要dp两个人分别先手的情况。
- 如果没有确定胜败就是平局。
- 需要处理的是，如果第二个人能走到一个必败态就确定是必胜态了，而第一个人所有出边都确定胜败了才知道必胜还是必败。
- 当然前面的部分做完了之后，如果第二个人某一个点，走不到必败态但能走到一个必胜态，在他的最优策略下他会直接走过去，这里需要进行另一个拓扑序dp。
- 后面这个dp只会变成使得第二个人必败，或者让第一个人因为所有出边都不可能平而必胜。

GAME ON GRAPH

- 类似的在反图拓扑序dp一下。
- 由于双方策略不同，所以需要dp两个人分别先手的情况。
- 如果没有确定胜败就是平局。
- 需要处理的是，如果第二个人能走到一个必败态就确定是必胜态了，而第一个人所有出边都确定胜败了才知道必胜还是必败。
- 当然前面的部分做完了之后，如果第二个人某一个点，走不到必败态但能走到一个必胜态，在他的最优策略下他会直接走过去，这里需要进行另一个拓扑序dp。
- 后面这个dp只会变成使得第二个人必败，或者让第一个人因为所有出边都不可能平而必胜。
- 所以任意顺序开始dp没有问题。

GAME ON GRAPH

- 类似的在反图拓扑序dp一下。
- 由于双方策略不同，所以需要dp两个人分别先手的情况。
- 如果没有确定胜败就是平局。
- 需要处理的是，如果第二个人能走到一个必败态就确定是必胜态了，而第一个人所有出边都确定胜败了才知道必胜还是必败。
- 当然前面的部分做完了之后，如果第二个人某一个点，走不到必败态但能走到一个必胜态，在他的最优策略下他会直接走过去，这里需要进行另一个拓扑序dp。
- 后面这个dp只会变成使得第二个人必败，或者让第一个人因为所有出边都不可能平而必胜。
- 所以任意顺序开始dp没有问题。
- 时间复杂度$O(n+m)$。

PEBBLE GAME

PEBBLE GAME

- 有一张有向图, Alice和Bob在进行一个游戏。

PEBBLE GAME

- 有一张有向图，Alice和Bob在进行一个游戏。
- 一开始Alice在$0 \cdots x-1$位置放一个棋子，Bob在$x \cdots n-1$位置放一个。

PEBBLE GAME

- 有一张有向图，Alice和Bob在进行一个游戏。
- 一开始Alice在$0 \cdots x-1$位置放一个棋子，Bob在$x \cdots n-1$位置放一个。
- 每次Alice移动一步，然后Bob移动一步，每一步不能移动到对方的棋子。如果无路可走，则立刻判负。

PEBBLE GAME

- 有一张有向图，Alice和Bob在进行一个游戏。
- 一开始Alice在$0 \cdots x-1$位置放一个棋子，Bob在$x \cdots n-1$位置放一个。
- 每次Alice移动一步，然后Bob移动一步，每一步不能移动到对方的棋子。如果无路可走，则立刻判负。
- 这个游戏中有很多小局：每一步后都会算出每个点到这两个棋子的最短距离，如果到两个棋子之一比另一个近，则近的这个棋子加一分，分数严格更多的棋子的拥有者赢下这一轮。

PEBBLE GAME

- 有一张有向图，Alice和Bob在进行一个游戏。
- 一开始Alice在$0 \cdots x-1$位置放一个棋子，Bob在$x \cdots n-1$位置放一个。
- 每次Alice移动一步，然后Bob移动一步，每一步不能移动到对方的棋子。如果无路可走，则立刻判负。
- 这个游戏中有很多小局：每一步后都会算出每个点到这两个棋子的最短距离，如果到两个棋子之一比另一个近，则近的这个棋子加一分，分数严格更多的棋子的拥有者赢下这一轮。
- 这个最短距离是无向图后的距离。

PEBBLE GAME

PEBBLE GAME

- 如果某个时刻后，双方的胜场数差达到了10^{100}，则分数高的获得游戏的胜利。

PEBBLE GAME

- 如果某个时刻后，双方的胜场数差达到了10^{100}，则分数高的获得游戏的胜利。
- 如果一直没发生两种结束方法之一，则双方平局。

PEBBLE GAME

- 如果某个时刻后，双方的胜场数差达到了10^{100}，则分数高的获得游戏的胜利。
- 如果一直没发生两种结束方法之一，则双方平局。
- 双方无限聪明。求对于$x \times (n-x)$种初始棋子放法中，有多少Alice获胜，有多少Bob获胜，有多少平局。

PEBBLE GAME

- 如果某个时刻后，双方的胜场数差达到了10^{100}，则分数高的获得游戏的胜利。
- 如果一直没发生两种结束方法之一，则双方平局。
- 双方无限聪明。求对于$x \times (n-x)$种初始棋子放法中，有多少Alice获胜，有多少Bob获胜，有多少平局。
- $n \le 50$。

PEBBLE GAME

- 如果某个时刻后，双方的胜场数差达到了10^{100}，则分数高的获得游戏的胜利。
- 如果一直没发生两种结束方法之一，则双方平局。
- 双方无限聪明。求对于$x \times (n-x)$种初始棋子放法中，有多少Alice获胜，有多少Bob获胜，有多少平局。
- $n \leq 50$。
- ~~实际上这题当场难度有~~$\geq 50\%$~~在于得到真题意。~~

PEBBLE GAME

PEBBLE GAME

- 好现在你得到了真题意。

PEBBLE GAME

- 好现在你得到了真题意。
- 首先你可以知道所有「两个棋子分别在x,y」的时候谁赢。

PEBBLE GAME

- 好现在你得到了真题意。
- 首先你可以知道所有「两个棋子分别在x,y」的时候谁赢。
- 现在先手一定希望胜场数和对面的差尽量大。

PEBBLE GAME

- 好现在你得到了真题意。
- 首先你可以知道所有「两个棋子分别在x,y」的时候谁赢。
- 现在先手一定希望胜场数和对面的差尽量大。
- 考虑`dp[k][x][y]`表示走k步，先手在x，后手在y，先手-后手的最大值。

PEBBLE GAME

- 好现在你得到了真题意。
- 首先你可以知道所有「两个棋子分别在x,y」的时候谁赢。
- 现在先手一定希望胜场数和对面的差尽量大。
- 考虑`dp[k][x][y]`表示走k步，先手在x，后手在y，先手-后手的最大值。
- 先不考虑走不了的问题，如果先手会赢，则这个`dp[k][x][y]`随着`k`增大会逐渐趋近于∞，如果后手会赢，会趋近于$-\infty$，如果平局会收敛或者在一个极小范围摆动，并且绝对值不会很大。

PEBBLE GAME

- 好现在你得到了真题意。
- 首先你可以知道所有「两个棋子分别在x,y」的时候谁赢。
- 现在先手一定希望胜场数和对面的差尽量大。
- 考虑`dp[k][x][y]`表示走k步，先手在x，后手在y，先手-后手的最大值。
- 先不考虑走不了的问题，如果先手会赢，则这个`dp[k][x][y]`随着`k`增大会逐渐趋近于∞，如果后手会赢，会趋近于$-\infty$，如果平局会收敛或者在一个极小范围摆动，并且绝对值不会很大。
- 如果先手走不了，那么直接是$-\infty$就行，因为已经强制输了。

PEBBLE GAME

PEBBLE GAME

- 考虑设一个比较大的k，然后用`dp[k][x][y]`和`dp[2k][x][y]`做对比，判一下是哪种情况。

PEBBLE GAME

- 考虑设一个比较大的k，然后用`dp[k][x][y]`和`dp[2k][x][y]`做对比，判一下是哪种情况。
- 瞎调调参就能过~~(可能是数据太水?)~~

PEBBLE GAME

- 考虑设一个比较大的k，然后用`dp[k][x][y]`和`dp[2k][x][y]`做对比，判一下是哪种情况。
- 瞎调调参就能过~~(可能是数据太水?)~~
- 时间复杂度$O(kn^3)$。

ZIGZAG GAME

ZIGZAG GAME

- 有个$n+n$的满二分图,每条边有**不同**的边权。

ZIGZAG GAME

- 有个$n+n$的满二分图，每条边有**不同**的边权。
- 开局Alice选择「升」或「减」中一个，Bob自动获得另一个。Alice选择把棋子放到一个点，然后Bob把棋子移动到相邻的一个点上。

ZIGZAG GAME

- 有个$n+n$的满二分图，每条边有**不同**的边权。
- 开局Alice选择「升」或「减」中一个，Bob自动获得另一个。Alice选择把棋子放到一个点，然后Bob把棋子移动到相邻的一个点上。
- Alice先开始，双方每次选一个相邻且没走过的点走过去，要求如果选的是减则走的边必须比上一条小，否则必须比上一条大，走不了的失败。

ZIGZAG GAME

- 有个$n+n$的满二分图，每条边有**不同**的边权。
- 开局Alice选择「升」或「减」中一个，Bob自动获得另一个。Alice选择把棋子放到一个点，然后Bob把棋子移动到相邻的一个点上。
- Alice先开始，双方每次选一个相邻且没走过的点走过去，要求如果选的是减则走的边必须比上一条小，否则必须比上一条大，走不了的失败。
- 交互题，边权给定，你需要选择Alice或Bob然后和交互库玩这个游戏，并战胜它。

ZIGZAG GAME

ZIGZAG GAME

- 事实上Bob必胜。

ZIGZAG GAME

- 事实上Bob必胜。
- 考虑构造一个必胜策略：设Alice选了升，然后一开始是x方点。

ZIGZAG GAME

- 事实上Bob必胜。
- 考虑构造一个必胜策略：设Alice选了升，然后一开始是x方点。
- 考虑构造一个匹配p，使得每次若现在在X_a就走到Y_{p_a}。

ZIGZAG GAME

- 事实上Bob必胜。
- 考虑构造一个必胜策略：设Alice选了升，然后一开始是x方点。
- 考虑构造一个匹配p，使得每次若现在在X_a就走到Y_{p_a}。
- 考虑一个做法：从大到小考虑每条边x, y，如果x, y都没匹配就直接匹配上。

ZIGZAG GAME

- 事实上Bob必胜。
- 考虑构造一个必胜策略：设Alice选了升，然后一开始是x方点。
- 考虑构造一个匹配p，使得每次若现在在X_a就走到Y_{p_a}。
- 考虑一个做法：从大到小考虑每条边x, y，如果x, y都没匹配就直接匹配上。
- 考虑这玩意的正确性：设$w_{1,p_1} < w_{2,p_2} < \cdots < w_{n,p_n}$，那么一条非匹配边$x, p_y$有以下情况：

ZIGZAG GAME

- 事实上Bob必胜。
- 考虑构造一个必胜策略:设Alice选了升,然后一开始是x方点。
- 考虑构造一个匹配p,使得每次若现在在X_a就走到Y_{p_a}。
- 考虑一个做法:从大到小考虑每条边x,y,如果x,y都没匹配就直接匹配上。
- 考虑这玩意的正确性:设$w_{1,p_1} < w_{2,p_2} < \cdots < w_{n,p_n}$,那么一条非匹配边$x,p_y$有以下情况:

1. $x < y$,那么$w_{x,p_y} < w_{y,p_y}$,显然Alice不可能从Y_{p_y}走到X_x。

ZIGZAG GAME

- 事实上Bob必胜。
- 考虑构造一个必胜策略：设Alice选了升，然后一开始是x方点。
- 考虑构造一个匹配p，使得每次若现在在X_a就走到Y_{p_a}。
- 考虑一个做法：从大到小考虑每条边x, y，如果x, y都没匹配就直接匹配上。
- 考虑这玩意的正确性：设$w_{1,p_1} < w_{2,p_2} < \cdots < w_{n,p_n}$，那么一条非匹配边$x, p_y$有以下情况：

1. $x < y$，那么$w_{x,p_y} < w_{y,p_y}$，显然Alice不可能从Y_{p_y}走到X_x。
2. $x > y$，那么才有可能$w_{x,p_y} > w_{y,p_y}$，在此情况下Alice可以从Y_{p_y}走到X_x，但是由1得$w_{x,p_y} < w_{x,p_x}$，就gg了，，，

ZIGZAG GAME

ZIGZAG GAME

- 我们考虑要的这个匹配有啥性质：不存在x, y使得$w_{x,p_x} < w_{y,p_x} < w_{y,p_y}$。

ZIGZAG GAME

- 我们考虑要的这个匹配有啥性质:不存在x, y使得$w_{x,p_x} < w_{y,p_x} < w_{y,p_y}$。
- 在此情况下y匹配p_x会更优。

ZIGZAG GAME

- 我们考虑要的这个匹配有啥性质：不存在x,y使得$w_{x,p_x} < w_{y,p_x} < w_{y,p_y}$。
- 在此情况下y匹配p_x会更优。
- 所以对于x,p_x的优先级是w_{x,p_x}越小越好，对于p_x是w_{x,p_x}越大越好。

ZIGZAG GAME

- 我们考虑要的这个匹配有啥性质：不存在x, y使得$w_{x,p_x} < w_{y,p_x} < w_{y,p_y}$。
- 在此情况下y匹配p_x会更优。
- 所以对于x, p_x的优先级是w_{x,p_x}越小越好，对于p_x是w_{x,p_x}越大越好。
- 然后在此基础上是个稳定婚姻问题，可以用$O(n^2)$的复杂度解决。

ZIGZAG GAME

ZIGZAG GAME

- 还记得稳定婚姻问题怎么做吗? ~~相信大家都记得，所以这一页没啥用了~~

ZIGZAG GAME

- 还记得稳定婚姻问题怎么做吗? ~~相信大家都记得，所以这一页没啥用子~~
- 初始每个X方点都没匹配，每次找一个没匹配的找到优先级最高的没试过的Y方点，如果没匹配就匹配，如果匹配过了，如果比这个Y方点原来匹配的更优就换。

ZIGZAG GAME

- 还记得稳定婚姻问题怎么做吗？~~相信大家都记得，所以这一页没啥用了~~
- 初始每个X方点都没匹配，每次找一个没匹配的找到优先级最高的没试过的Y方点，如果没匹配就匹配，如果匹配过了，如果比这个Y方点原来匹配的更优就换。
- 拿个队列或栈存没匹配的点，显然总共会问n^2次，时间复杂度$O(n^2)$。

THANKS!