

I

找规律

1	0	1
3	5	6
10	5	15
6	3	5

From:2016西电微软俱乐部招新选拔题

\wedge xor

给你n个数，其中有且仅有一个数出现了奇数次，其余的数都出现了偶数次。用线性时间常数空间找出出现了奇数次的那一个数。

给你 n 个数，其中有且仅有两个数出现了奇数次，其余的数都出现了偶数次。用线性时间常数空间找出出现了奇数次的那两个数。

D.....? P.....?

dynamic programming is a method for solving a complex problem by **breaking it down into a collection of simpler subproblems**.

Sublimation

Room 201

msc_sublimation@outlook.com

qq

有 n 堆石子，放在 n 个台阶上，每次可以从上面的台阶取任意多个石子放到下一级台阶，或者将第一级台阶的石子直接取走。

如何操作？

N 1000

$A[i]$ 1000

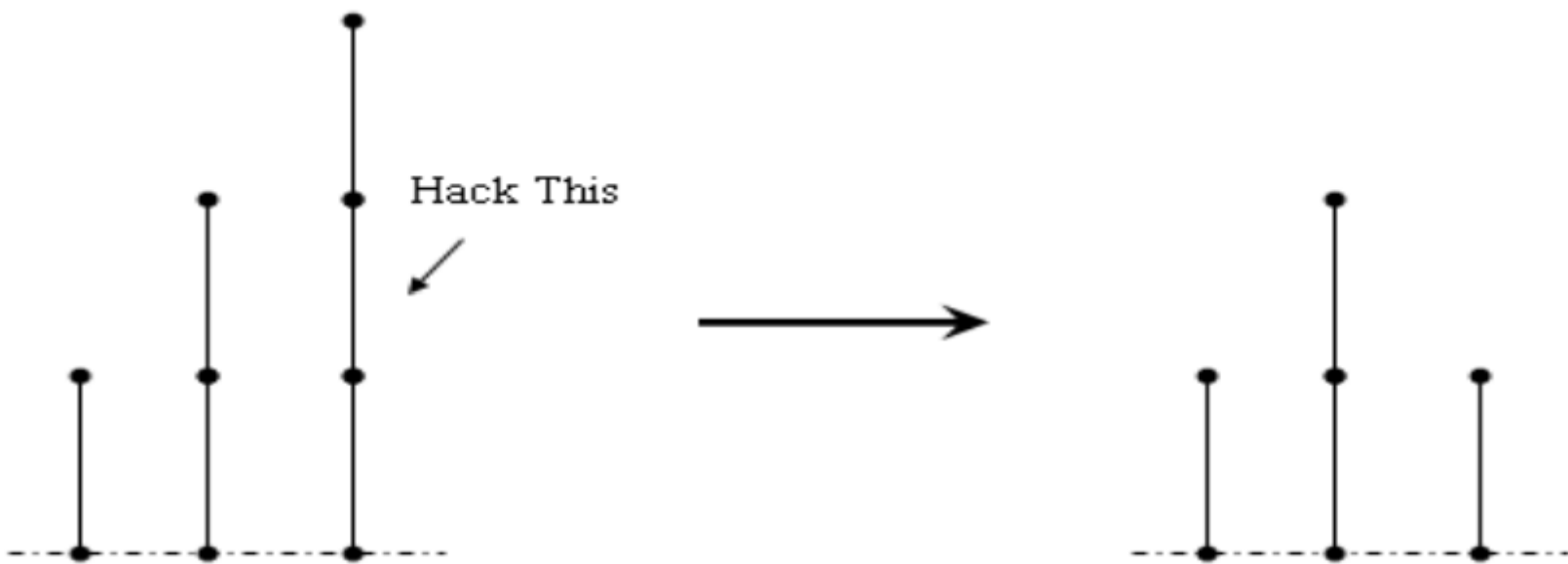
International chess

ju

车

王后

马?



海盜分金

Game theory

组合游戏

- 游戏有 2 名参与者，通常称为玩家 L 和玩家 R。
- 游戏过程中的任意时刻有确定的状态。
- 参与者操作时将游戏从当前状态转移到另一状态，规则规定了在任意一个状态时，参与者可以到达的状态集合。
- 参与者轮流进行操作。
- 在游戏处于某状态，当前参与者不能进行操作时，游戏结束。本节只讨论最先不能进行操作者输的情况。
- 无论参与者做出怎样的操作，游戏总会在有限步数之内结束(没有平局)。
- 参与者拥有游戏本身，和游戏过程的所有信息，比如规则、自己和对手之前的操作。

无偏博弈 sg 函数

不平等博弈 Surreal Number

状态

必胜

必败

2幂次博弈

代码:

有一个01的矩阵 $n*m$ $n,m \leq 1000$

两人轮流操作；

每次选择一个数值为1的点；

使得每次选择的点到右下角的矩阵01状态翻转；

无法操作则输！

问先手胜后首胜。

```
2
2 2
1 1
1 1
3 3
0 0 0
0 0 0
0 0 0
```

hdu5963

一棵树，每个边有一个边权1,0

每次选一个边权为1的边

将从这个边开始到根的路径的所有边的边权
翻转。

不能操作则输。

操作：

- 1.更改一个边的边权
- 2.查询以某一个点为根的子树先手必胜必败？

$N \leq 1e5$

操作数 $\leq 1e5$

Tree

套路！

- 定义一种新的黑白棋：
- 1. 棋盘大小为 $5*5$ 的格子；
- 2. 有些格子不能放棋子；
- 3. 同一个格子最多放一个棋子；
- 4. 先手执白棋，后手执黑棋；
- 5. 先手第一次可以把棋放在任意可以放的位置上；
- 6. 接下来两人轮流放棋子，这个棋子必须与上一个人放的棋子相邻
- 请问：两人都是最优策略，是先手赢，还是先手输？

- 输入
- 有多组输入数据，第一行为一个数字T，代表有T组输入数据 ($0 < T \leq 10$)。
- 接下来为T组数据。
- 每组数据分5行、每行5个数字构成，每个数字为0或1。0表示这个位置可以放棋子，1表示这个位置不能放棋子。
- 输出
- 对于每组数据，在一行上输出“win”或“lose”，表示先手赢或输。

- 样例输入

2

11111 11111

11111 11111

11111 11111

11111 11111

00000 10000

样例输出

win lose

有一堆石子双方每轮可以取1~4个

多堆？

NIM

有 n 堆石子，每次可以从任意一堆取走任意多个石子。取不了则输。

如何操作？

$N \ 1000$

$A[i] \ 1000$

International chess

ju
车

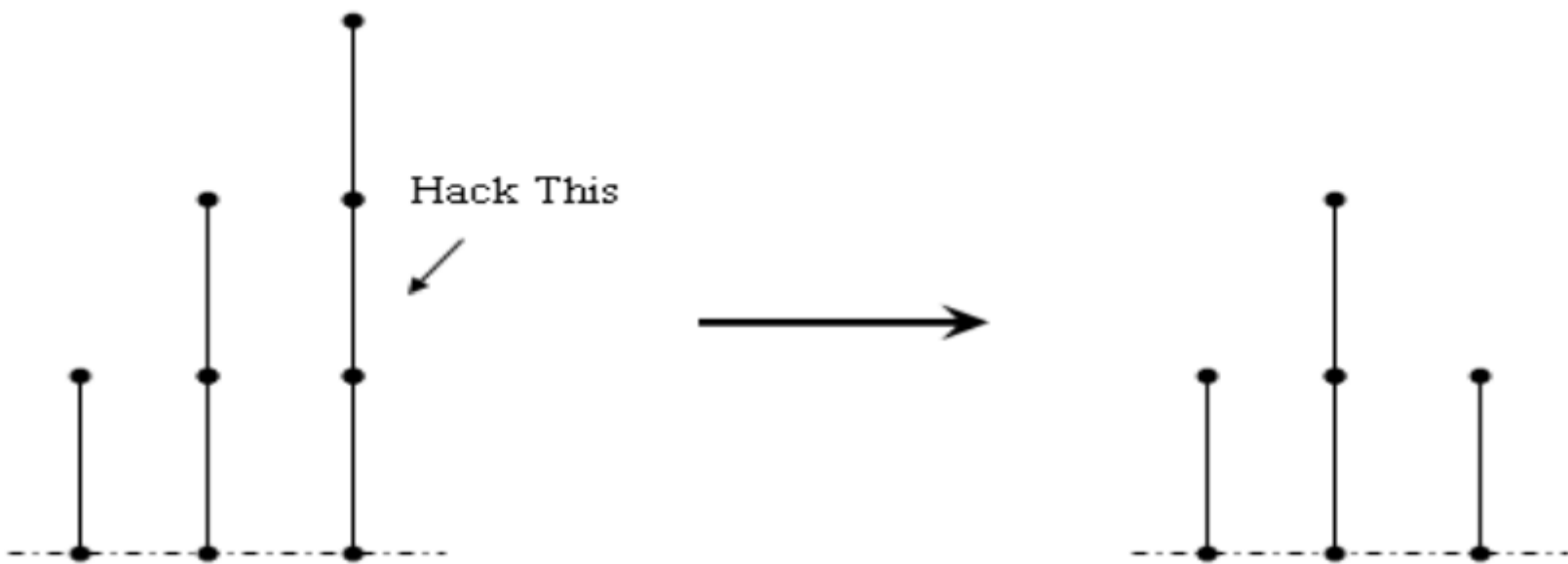
马?

有 n 堆石子，放在 n 个台阶上，每次可以从上面的台阶取任意多个石子放到下一级台阶，或者将第一级台阶的石子直接取走。

如何操作？

N 1000

$A[i]$ 1000



$$SG(x) = \begin{cases} 0(x \text{ is a winning end position}) \\ \text{mex}\{SG(y) \mid y \in f(x)\}(\text{other situation}) \end{cases}$$

通式动态规划解法

- 步骤1: 把所有“胜利终止状态”标记为P状态, “失败终止状态”标记为N状态。
 - 步骤2: 找到所有的未定状态中, 所有后继都被确定是N状态的状态, 设置为P状态。
 - 步骤3: 找到所有的未定状态中, 可以一步到达P状态的状态, 都设置为N状态。
 - 步骤4: 若上两步中没有产生新的P状态或N状态, 程序结束, 否则回到步骤2。
-
- 时间复杂度——所有状态的决策数之和

游戏

- 甲乙两人面对若干排石子。
- 每一排石子的数目可以任意确定。
- 两人轮流按下列规则取走一些石子：
 - 每一步必须从某一排中取走一或两枚石子；
 - 这两枚石子必须是紧紧挨着的；
 - 如果谁无法按规则取子，谁就是输家。

$a_1=7$ 1 2 3 4 5 6 7
○ ○ ○ ○ ○ ○ ○

$a_2=3$ ○ ○ ○

$a_3=5$ ○ ○ ○ ○ ○

有一个整数 S (≥ 2)，先行者在 S 上减掉一个数 x ，至少是1，但小于 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：谁有必胜策论。

k倍动态减法游戏

有一个整数 S (≥ 2)，先行者在 S 上减掉一个数 x ，至少是1，但小于 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：谁有必胜策论。

A第一回合減去2

A获胜B第一回合減去4

A第二回合減去1

K=2

通式解法

- $NP(m,n)$ 表示S还剩下m且接下去即将操作的玩家最多能减去n的状态，则初始状态为 $NP(S,S-1)$ 。
- 规定，若在 $NP(m,n)$ 状态下，即将操作的玩家必胜则 $NP(m,n)=1$ ，否则 $NP(m,n)=0$ 。
- 若用动态规划计算所有 $NP(m,n)$ ，则判定胜负的时间复杂度为 $O(n^3)$ 。

优化1

状态单调性



**状态NP(m,n)是关于
关于n单调不减的。**

$$\text{记 } f(m) = \min\{n \mid NP(m, n) = 1\}$$

1																
0	1															
1	1	1														
0	0	0	1													
1	1	1	1	1												
0	1	1	1	1	1											
1	1	1	1	1	1	1										
0	0	0	0	0	0	0	1									
1	1	1	1	1	1	1	1	1								
0	1	1	1	1	1	1	1	1	1							
1	1	1	1	1	1	1	1	1	1	1						
0	0	0	1	1	1	1	1	1	1	1	1					
1	1	1	1	1	1	1	1	1	1	1	1	1				
0	1	1	1	1	1	1	1	1	1	1	1	1	1			
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

图4-1 $k=1$ 时 $NP(m, n)$ 的情况 红色格表示 $f(m)$

优化1

若 $n_0 = f(m)$,
 $NP(m, n_0) = 0$,

则

若且仅当 $f(m - n_0) > kn_0$ 且 $f(m - (n_0 - 1)) \leq k(n_0 - 1)$ 。

则

对于任意 $r = 1, 2, 3, \dots, n$ 有 $m - r \geq 0$ 且 $NP(m - r, kr) = 1$ 。
 动态转移方程: $NP(m, n) = 0$ 且 $NP(m - n_0 + 1, k(n_0 - 1)) = 1$

时间复杂度: $O(S^2)$

优化2-决策单调性

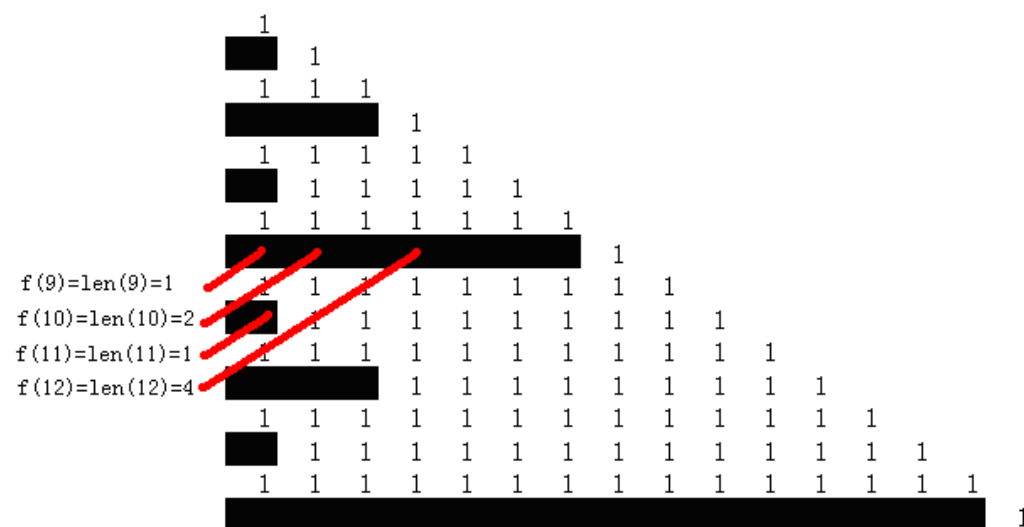


图4-2 $k=1$ 时 $NP(m, n)$ 的情况 $NP(m, n)=0$ 的连续的格子就像一堵堵墙

优化2—决策单调性

- 所有这些直线是平行的
- 随着 m 增大逐渐向下向右移
- 每一堵墙都是固定的、右端有界的



用栈储存“墙”

优化2—决策单调性

- 逐个检验栈中的“墙”
- 若某堵“墙”不能挡住从 $(m,0)$ 格子出发斜率为 k^{-1} 的直线，那么该“墙”出栈
- 否则，若这堵“墙”能挡住斜线，则循环结束并得出 $f(m)$ 的值。
- 最后，根据 $f(m)$ 可确定一堵新“墙”的位置和长度，新“墙”入栈。
- 时间复杂度： $O(S)$

BOI 2008 game

- 一个 $n*n$ 的棋盘，每个格子要么是黑色要么是白色。白格子是游戏区域，黑格子表示障碍。
- 指定两个格子**A****B**，分别是先手方和后手方的起始格子。**A**和**B**这两格子不重合。
- 游戏中，双方轮流操作。每次操作，玩家向上下左右四个格子之一走一步，但不能走进黑色格子。有一种特殊情况，当一方玩家，恰好走到当前对方所在的格子里，他就可以再走一步（不必是同一方向），“跳过对手”。
- 胜负的判定是这样的，若有一方走进对方的起始格子，就算获胜，即使是跳过对方，也算获胜。

I