

由感性认识到理性认识

——透析一类博弈游戏的解答过程

一、 游戏..... 2

二、 从简单入手..... 2

三、 类比与联想..... 6

四、 证明..... 8

五、 推广..... 11

六、 精华..... 12

七、 结论..... 16

八、 总结..... 17

一、游戏

📖 游戏 A:

📖 甲乙两人面对若干堆石子，其中每一堆石子的数目可以任意确定。例如图 1 所示的初始局面：共 $n=3$ 堆，其中第一堆的石子数 $a_1=3$ ，第二堆石子数 $a_2=3$ ，第三堆石子数 $a_3=1$ 。两人轮流按下列规则取走一些石子，游戏的规则如下：

- 每一步应取走至少一枚石子；
- 每一步只能从某一堆中取走部分或全部石子；
- 如果谁无法按规则取子，谁就是输家。

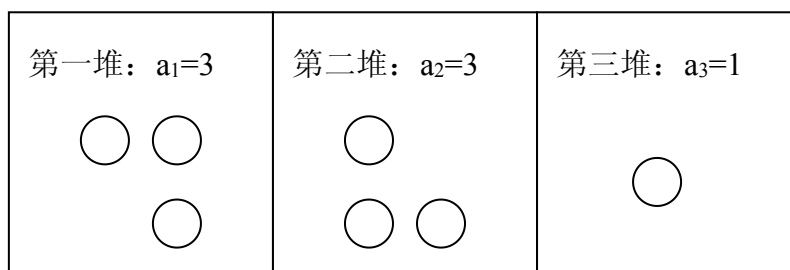


图 1 游戏的一个初始局面

📖 游戏 B:

- 甲乙双方事先约定一个数 m ，并且每次取石子的数目不能超过 m 个；
- 其余规则同游戏 A。

我们关心的是，对于一个初始局面，究竟是先行者（甲）有必胜策略，还是后行者（乙）有必胜策略。

下面，我们从简单入手，先来研究研究这个游戏的一些性质。

二、从简单入手

👉 用一个 n 元组 (a_1, a_2, \dots, a_n) ，来描述游戏过程中的一个局面。

👉 可以用 3 元组 $(3, 3, 1)$ 来描述图 1 所示的局面。

✎ 改变这个 n 元组中数的顺序，仍然代表同一个局面。

👉 $(3, 3, 1)$ 和 $(1, 3, 3)$ ，可以看作是同一个局面。

✎ 如果初始局面只有一堆石子，则甲有必胜策略。

📖 甲可以一次把这一堆石子全部取完，这样乙就无石子可取了。

✎ 如果初始局面有两堆石子，而且这两堆石子的数目相等，则乙有必胜策略。

📖 因为有两堆石子，所以甲无法一次取完；

📖 如果甲在一堆中取若干石子，乙便在另一堆中取同样数目的石子；

📖 根据对称性，在甲取了石子之后，乙总有石子可取；

📖 石子总数一直在减少，最后必定是甲无石子可取。

👉 对于初始局面 (1) ，甲有必胜策略，而初始局面 $(3, 3)$ ，乙有必胜策略。

👉 局面的加法： $(a_1, a_2, \dots, a_n) + (b_1, b_2, \dots, b_m) = (a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_m)$ 。

👉 $(3) + (3) + (1) = (3, 3) + (1) = (3, 3, 1)$ 。

👉 对于局面 A, B, S ，若 $S=A+B$ ，则称局面 S 可以分解为“子局面” A 和 B 。

👉 局面 $(3, 3, 1)$ 可以分解为 $(3, 3)$ 和 (1) 。

✎ 如果初始局面可以分成两个相同的“子局面”，则乙有必胜策略。

📖 设初始局面 $S=A+A$ ，想象有两个桌子，每个桌子上放一个 A 局面；

📖 若甲在一个桌子中取石子，则乙在另一个桌子中对称的取石子；

📖 根据对称性，在甲取了石子之后，乙总有石子可取；

📖 石子总数一直在减少，最后必定是甲无石子可取。

👉 初始局面 $(2, 2, 5, 5, 5, 5, 7, 7)$ ，可以分成两个 $(2, 5, 5, 7)$ ，故乙有必胜策略。

☞ 对于局面 S ，若先行者有必胜策略，则称“ S 胜”。

☞ 对于局面 S ，若后行者有必胜策略，则称“ S 负”。

☞ 若 $A=(1)$ ， $B=(3, 3)$ ， $C=(2, 2, 5, 5, 5, 5, 7, 7)$ ，则 A 胜， B 负， C 负。

☞ 我们所关心的，就是如何判断局面的胜负。

✎ 如果局面 S 胜，则必存在取子的方法 $S \rightarrow T$ ，且 T 负。

✎ 如果局面 S 负，则对于任意取子方法 $S \rightarrow T$ ，有 T 胜。

☞ 设初始局面 S 可以分解成两个子局面 A 和 B （分解理论）。

✎ 若 A 和 B 一胜一负，则 S 胜。

☞ 不妨设 A 胜 B 负；

☞ 想象有两个桌子 A 和 B ，桌子上分别放着 A 局面和 B 局面；

☞ 因为 A 胜，所以甲可以保证取桌子 A 上的最后一个石子；

☞ 与此同时，甲还可以保证在桌子 B 中走第一步的是乙；

☞ 因为 B 负，所以甲还可以保证取桌子 B 中的最后一个石子；

☞ 综上所述，甲可以保证两个桌子上的最后一个石子都由自己取得。

✎ 若 A 负 B 负，则 S 负。

☞ 无论甲先从 A 中取，还是先从 B 中取，都会变成一胜一负的局面；

☞ 因此，乙面临的局面总是“胜”局面，故甲面临的 S 是“负”局面。

✎ 若 B 负，则 S 的胜负情况与 A 的胜负情况相同。

✎ 若 A 胜 B 胜，则有时 S 胜，有时 S 负。

✎ 如果 $S=A+C+C$ ，则 S 的胜负情况与 A 相同。

📖 令 $B=C+C$ ，则 $S=A+B$ 且 B 负，故 S 的胜负情况与 A 相同。

👉 图 1 所示的初始局面 $(3, 3, 1) = (3) + (3) + (1)$ ，与局面(1)的胜负情况相同。

👉 图 1 中所示的初始局面 $(3, 3, 1)$ 是“胜”局面，甲有必胜策略。

👉 称一个石子也没有的局面为“空局面”。

✎ 空局面是“负”局面。

👉 如果局面 S 中，存在两堆石子，它们的数目相等。用 T 表示从 S 中把这两堆石子拿掉之后的局面，则称“ S 可以简化为 T ”。

👉 局面 $(2, 2, 2, 7, 9, 9)$ 可以简化为 $(2, 2, 2, 7)$ ，还可以进一步简化为 $(2, 7)$ 。

✎ 一个局面的胜负情况，与其简化后的局面相同。

👉 三个局面 $(2, 2, 2, 7, 9, 9)$ 、 $(2, 2, 2, 7)$ 和 $(2, 7)$ ，胜负情况都相同。

👉 不能简化的局面称为“最简局面”。

👉 局面 $(2, 7)$ 是最简局面。

✎ 最简局面中不会有两堆相同的石子，故可以用一个集合来表示最简局面。

👉 最简局面 $(2, 7)$ 可以用集合 $\{2, 7\}$ 来表示。

✉ 如果只关心局面的胜负，则一个局面可以用一个集合来描述。

👉 图 1 所示的局面 $(3, 3, 1)$ ，可以用集合 $\{1\}$ 来描述。

如果用搜索（博弈树）的方法来解决这个游戏，则采用集合来表示一个局面，比采用多元组来表示一个局面，搜索量将有所减少，但时间复杂度仍然很高。

能不能进一步简化一个局面的表示呢？

三、类比与联想

☞ 二进制加法¹

- $1 + 0 = 1$;
- $0 + 1 = 1$;
- $0 + 0 = 0$;
- $1 + 1 = 0$ 。

☞ 二进制的加法 VS 局面的加法

- 大写字母 AB 表示局面，小写字母 ab 表示二进制
- 若 A 和 B 相同，则 A+B 负；若 a 和 b 相等，则 $a+b=0$
- 若 A 胜 B 负，则 A+B 胜；若 $a=1$ 且 $b=0$ ，则 $a+b=1$
- 若 B 胜 A 负，则 A+B 胜；若 $b=1$ 且 $a=0$ ，则 $a+b=1$
- 若 A 负 B 负，则 A+B 负；若 $a=0$ 且 $b=0$ ，则 $a+b=0$
-

★ 如果用二进制 1 和 0，分别表示一个局面的胜或负

✎ 局面的加法，与二进制的加法有很多类似之处。

✖ 若 A 胜 B 胜，则 A+B 有时胜，有时负；若 $a=1$ 且 $b=1$ ，则 $a+b=0$ 。

¹ 本文的“二进制加法”，是指不进位的二进制加法，也可以理解为逻辑里的“异或”操作。

☞ 二进制数的加法：对二进制数的每一位，都采用二进制的加法。

$$\begin{array}{r} 0011 \\ + 1010 \\ \hline 1001 \end{array}, \begin{array}{r} 1010 \\ + 1010 \\ \hline 0000 \end{array}。$$

☞ 二进制数的加法 VS 局面的加法

- 大写字母 AB 表示局面，小写字母 ab 表示二进制数
- 若 A 和 B 相同，则 A+B 负；若 a 和 b 相等，则 a+b 为 0
- 若 A 胜 B 负，则 A+B 胜；若 $a \neq 0$ 且 $b=0$ ，则 $a+b \neq 0$
- 若 B 胜 A 负，则 A+B 胜；若 $b \neq 0$ 且 $a=0$ ，则 $a+b \neq 0$
- 若 A 负 B 负，则 A+B 负；若 $a=0$ 且 $b=0$ ，则 $a+b=0$
- 若 A 胜 B 胜，则 A+B 有时胜，有时负
- 若 $a \neq 0$ 且 $b \neq 0$ ，则有时 $a+b \neq 0$ ，有时 $a+b=0$
-

☞ 如果用二进制数 s 来表示一个局面 S 的胜或负，S 胜则 $s \neq 0$ ，S 负则 $s=0$

☞ 局面的加法，与二进制数的加法，性质完全相同。

☞ 能否用一个二进制数，来表示一个局面呢？

☞ 用符号 #S，表示局面 S 所对应的二进制数。

☞ 如果局面 S 只有一堆石子，则用这一堆石子数目所对应的二进制数来表示 S。

☞ $\#(5)=5=101。$

★ 若局面 $S=A+B$ ，则 $\#S=\#A+\#B$ 。

👉 局面 $(3, 3)=(3)+(3)$ ，所以 $\#(3, 3)=\#(3)+\#(3)=11+11=0$ 。

👉 局面 $(3, 3, 1)=(3, 3)+(1)$ ，所以 $\#(3, 3, 1)=\#(3, 3)+\#(1)=0+1=1$ 。

👉 函数 f ：若局面 S 只有一堆石子，设 $S=\{a_1\}$ ，则 $f(a_1)=\#S$ ，即 $f(a_1)=\#(a_1)$ 。

👉 对于游戏 A 来说， $\#(5)=101$ ，所以 $f(5)=101$ 。

👉 对于游戏 A 来说， $f(x)$ 就是 x 所对应的二进制数。换句话说， $f(x)=x$ 。

👉 设局面 $S=(a_1, a_2, \dots, a_n)$ ，即 $S=(a_1)+(a_2)+\dots+(a_n)$ ，则 $\#S=f(a_1)+f(a_2)+\dots+f(a_n)$ 。

👉 $\#(3, 3, 1)=\#((3)+(3)+(1))=\#(3)+\#(3)+\#(1)=f(3)+f(3)+f(1)=11+11+1=1$ 。

★ 对于局面 S ，若 $\#S=0$ ，则 S 负；若 $\#S \neq 0$ ，则 S 胜。

四、证明

👉 二进制数 a, b ，若 $a + b = 0$ ，当且仅当 $a = b$ 。

👉

| | |
|--------|--------|
| 1011 | 1011 |
| + 1011 | + 1001 |
| 0000 | 0010 |

👉 二进制数 a, b, s ，若 $a + b = s$ ，则 $a = b + s$ 。

👉

| | |
|--------|--------|
| 0011 | 1001 |
| + 1010 | + 1010 |
| 1001 | 0011 |

✎ 二进制数 $a_1+a_2+\dots+a_n=p \neq 0$ ，则必存在 k ，使得 $a_k+p < a_k$ 。

📖 因为 $p \neq 0$ ，所以 p 的最高位是 1；

📖 设 p 的最高位是第 q 位；

📖 至少存在一个 k ，使得 a_k 的第 q 位也是 1；

📖 a_k+p 的第 q 位为 0，所以 $a_k+p < a_k$ 。

$$\begin{array}{rcl}
 & 11001 & a_1 \\
 & 01101 & a_k \\
 + & 10010 & a_3 \\
 \hline
 & 00110 & p \\
 & q &
 \end{array}
 \qquad
 \begin{array}{rcl}
 & 01101 & a_k \\
 + & 00110 & p \\
 \hline
 & 01011 & a_k+p \\
 & q &
 \end{array}$$

👉

✎ 若 $\#S=0$ ，则无论先行者如何取子 $S \rightarrow T$ ，都有 $\#T \neq 0$ 。

📖 先行者只能从某一堆中取若干石子，不妨设他选择的的就是第 1 堆；

📖 设先行者从第 1 堆中取了 x 个石子，用 T 表示取完之后的局面；

📖 设 $S=(a_1, a_2, \dots, a_n)$ ，则 $T=(a_1-x, a_2, \dots, a_n)$ ；

📖 $\#S=f(a_1)+\#(a_2, \dots, a_n)=0$ ，故 $f(a_1)=\#(a_2, \dots, a_n)$ ；

📖 $\#T=f(a_1-x)+\#(a_2, \dots, a_n)=f(a_1-x)+f(a_1)$ ；

📖 $x>0 \rightarrow f(a_1) \neq f(a_1-x) \rightarrow f(a_1)+f(a_1-x) \neq 0 \rightarrow \#T \neq 0$ 。

$$\begin{array}{rcl}
 00101 & a_1 & 00101 & a_1 & x \neq a_1 \\
 10011 & a_2 & & & \\
 10111 & a_3 & 00101 & a_2+a_3+a_4=a_1 & a_1 \\
 + & 00001 & a_4 & + & + \\
 \hline
 00000 & p=0 & 00000 & p=0 & p \neq 0
 \end{array}$$

👉

✎ 若 $\#S \neq 0$ ，则先行者必然存在一种取子方法 $S \rightarrow T$ ，且 $\#T = 0$ 。

📖 设 $S = (a_1, a_2, \dots, a_n)$ ， $p = \#S = f(a_1) + f(a_2) + \dots + f(a_n)$ ；

📖 因为 $p \neq 0$ ，所以必然存在 k ，使得 $f(a_k) + p < f(a_k)$ ，不妨设 $k=1$ ， $f(a_1) + p = x$ ；

📖 先行者将第 1 堆的石子的数目从 a_1 变成 x ，用 T 表示这个局面；

📖 $p = \#S = f(a_1) + \#(a_2, \dots, a_n)$ ，故 $\#(a_2, \dots, a_n) = f(a_1) + p = x$ ；

📖 $\#T = f(x) + \#(a_2, \dots, a_n) = f(x) + x = 0$ 。

| | | | | | | |
|---------|------------|---|-------|---------------------------------------|---|---------|
| 00101 | a_1 | | 00101 | a_1 | | x |
| 10011 | a_2 | | | | | |
| 00111 | a_3 | | 00011 | $x = a_2 + a_3 + a_4 = a_1 + p < a_1$ | | x |
| + 10111 | a_4 | + | | | + | |
| 00110 | $p \neq 0$ | | 00110 | p | | $p = 0$ |



☞ 若 S 是空局面，则 $\#S = 0$ 。

✉ 若 $\#S = 0$ ，则 S 负；若 $\#S \neq 0$ ，则 S 胜。

👉 $\#(1, 2, 3) = 01 + 10 + 11 = 0$ ，故局面 $(1, 2, 3)$ 负。

👉 $\#(1, 2, 3, 4) = 001 + 010 + 011 + 100 = 100$ ，故局面 $(1, 2, 3, 4)$ 胜。

对于游戏 A 来说，任意的一个初始局面 $S = (a_1, a_2, \dots, a_n)$ ，我们把这里的 a_i 都看成是二进制数。令 $\#S = a_1 + a_2 + \dots + a_n$ 。若 $\#S \neq 0$ ，则先行者（甲）有必胜策略；否则 $\#S = 0$ ，这时后行者（乙）有必胜策略。

下面把这个结论推广到游戏 B。

☞ 函数 $f: f(x)=x \bmod (m+1)$; 把函数 f 的值看作是二进制数。

☞ 对于任意初始局面 $S=(a_1, a_2, \dots, a_n)$, 令 $\#S=f(a_1)+f(a_2)+\dots+f(a_n)$ 。

✎ 若 $\#S \neq 0$, 则先行者 (甲) 有必胜策略; 否则后行者 (乙) 有必胜策略。

📖 类似游戏 A 的证明。

📖 游戏 B 的解法与游戏 A 十分类似。这是因为两个游戏的规则相当类似。

五、推广

📖 游戏 C:

📖 甲乙两人面对若干排石子, 其中每一排石子的数目可以任意确定。例如图 2 所示的初始局面: 共 $n=3$ 排, 其中第一排的石子数 $a_1=7$, 第二排石子数 $a_2=3$, 第三排石子数 $a_3=3$ 。两人轮流按下列规则取走一些石子, 游戏的规则如下:

- 每一步必须从某一排中取走两枚石子;
- 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。

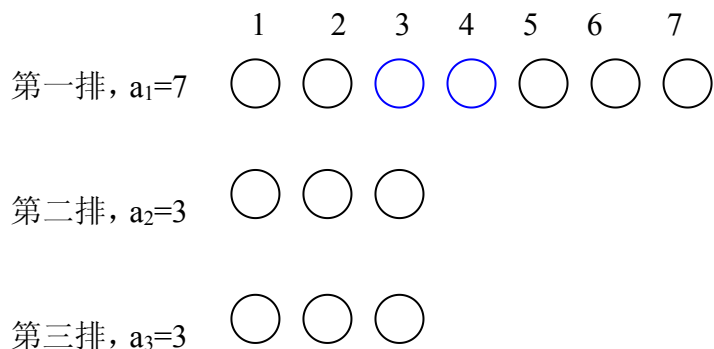


图 2 游戏的一个初始局面

☞ 如果甲第一步选择取第一排 34 这两枚石子, 之后无论是甲还是乙, 都不能一次取走 25 这两枚石子。换句话说, 如果取了 34 这两枚石子, 等价于将第一排分成了两排, 这两排分别有 2 个和 3 个石子。

我们只关心，对于一个初始局面，究竟是先行者（甲）有必胜策略，还是后行者（乙）有必胜策略。

游戏 C 的规则和游戏 A 并不那么相似。但是，前面所列出的，游戏 A 的关键性质，游戏 C 却都具有。比如说，图 2 所示的初始局面可以用三元组(7, 3, 3)来表示，它的胜负情况与初始局面(7)相同。

游戏 A 的解答是由它的性质得出来的。因此，我们猜想游戏 C 是否也能用类似的方法来解。

六、精华

☞ 回忆游戏 A 的结论，以及它在游戏 B 上的推广，对于游戏 C，我们的想法是

★ 设计一个函数 f ，把函数 f 的值看作是二进制数。对于任意一个初始局面 S ，设 $S=(a_1, a_2, \dots, a_n)$ ，令 $\#S=f(a_1)+f(a_2)+\dots+f(a_n)$ 。若 $\#S \neq 0$ ，则先行者（甲）有必胜策略；否则 $\#S=0$ ，这时后行者（乙）有必胜策略。

☞ 游戏 A 中， $f(x) = x$ 。

☞ 游戏 B 中， $f(x) = x \bmod (m+1)$ 。

☞ 游戏 C 中， $f(x) = ?$ 。

☞ 关键就在于如何构造一个满足要求的函数 f 。

☞ 回忆关于游戏 A、B 的结论的证明过程

✎ 函数 f 是否满足要求，关键在于 $\#S$ 是否满足下面的条件。

➤ 若 $\#S=0$ ，则无论先行者如何取子 $S \rightarrow T$ ，都有 $\#T \neq 0$ 。

➤ 若 $\#S \neq 0$ ，则先行者必然存在一种取子方法 $S \rightarrow T$ ，且 $\#T=0$ 。

$$\begin{array}{rcl}
 00101 & f(a_1) & 00101 & f(a_1) & \#(b_1, b_2, \dots, b_m) \in g(a_1) \\
 10011 & f(a_2) & & & \\
 10111 & f(a_3) & 00101 & f(a_1) & f(a_1) \\
 + & 00001 & f(a_4) & + & + \\
 \hline
 00000 & p=0 & 00000 & p=0 & p \neq 0
 \end{array}$$

↑

➤ 若 $\#S \neq 0$ ，则先行者必然存在一种取子方法 $S \rightarrow T$ ，且 $\#T=0$ 。

📖 设 $S=(a_1, a_2, \dots, a_n)$ ， $p=\#S=f(a_1)+f(a_2)+\dots+f(a_n)$ ；

📖 因为 $p \neq 0$ ，所以必然存在 k ，使得 $f(a_k)+p < f(a_k)$ ，不妨设 $k=1$ ， $f(a_1)+p=x$ ；

📖 因为 $p=\#S=f(a_1)+\#(a_2, \dots, a_n)$ ，故 $\#(a_2, \dots, a_n)=p+f(a_1)=x$ ；

📖 如果先行者把局面 (a_1) 变为局面 (b_1, \dots, b_m) ， $\#(b_1, \dots, b_m)$ 属于集合 $g(a_1)$ ；

📖 设这时的局面为 T ，我们有 $T=(b_1, \dots, b_m)+(a_2, \dots, a_n)$ ；

📖 $\#T=\#(b_1, \dots, b_m)+\#(a_2, \dots, a_n)=\#(b_1, \dots, b_m)+x$ ；

📖 如果要使 $\#T=0$ ，相当于要找到 (b_1, \dots, b_m) ，使得 $\#(b_1, \dots, b_m)$ 等于 x ；

📖 如果可以保证 x 属于集合 $g(a_1)$ ，则肯定可以找到相应的 (b_1, \dots, b_m) ；

📖 因为 $x < f(a_1)$ ，所以， x 属于集合 $\{0, 1, \dots, f(a_1)-1\}$ ；

📖 如果集合 $g(a_1)$ 包含集合 $\{0, 1, \dots, f(a_1)-1\}$ ，则 x 一定属于 $g(a_1)$ 。（充分）

$$\begin{array}{rcl}
 00101 & f(a_1) & 00101 & f(a_1) & \#(b_1, b_2, \dots, b_m)=x \\
 10011 & f(a_2) & & & \\
 00111 & f(a_3) & 00011 & x=f(a_1)+p < f(a_1) & x \quad \text{如果 } x \in g(a_1) \\
 + & 10111 & f(a_4) & + & + \\
 \hline
 00110 & p \neq 0 & 00110 & p & p=0
 \end{array}$$

↑

函数 f 满足要求的一个充分条件

- $f(a_1)$ 不属于集合 $g(a_1)$ 。
- 集合 $g(a_1)$ 包含集合 $\{0, 1, \dots, f(a_1)-1\}$ 。

👉 如果 $g(a_1)=\{0, 1, 2, 5, 7, 8, 9\}$ ，则 $f(a_1)=3$ ，满足要求。

☞ 用大写字母 N 表示非负整数集，即 $N=\{0, 1, 2, \dots\}$ 。

☞ 令 N 为全集，集合 $G(x)$ 表示集合 $g(x)$ 的补集。

☞ 定义函数 $f(n)$ ： $f(n)=\min\{G(n)\}$ ，即 $f(n)$ 等于集合 $G(n)$ 中的最小数。

☞ 设局面 $S=(a_1, a_2, \dots, a_n)$ ， $\#S=f(a_1)+f(a_2)+\dots+f(a_n)$ ，采用二进制数的加法。

☒ 若 $\#S=0$ ，则 S 负；若 $\#S \neq 0$ ，则 S 胜。

👉 游戏 C 的 f 值：

- $g(0)=\{\}$ ， $G(0)=\{0, 1, \dots\}$ ， $f(0)=0$ ；
- $g(1)=\{\}$ ， $G(1)=\{0, 1, \dots\}$ ， $f(1)=0$ ；
- $g(2)=\{\#(0)\}=\{f(0)\}=\{0\}$ ， $G(2)=\{1, 2, \dots\}$ ， $f(2)=1$ ；
- $g(3)=\{\#(1)\}=\{f(1)\}=\{0\}$ ， $G(2)=\{1, 2, \dots\}$ ， $f(3)=1$ ；
- $g(4)=\{\#(2), \#(1, 1)\}=\{f(2), f(1)+f(1)\}=\{1, 0\}$ ， $G(4)=\{2, 3, \dots\}$ ， $f(4)=2$ ；
- $g(5)=\{\#(3), \#(1, 2)\}=\{f(3), f(1)+f(2)\}=\{1, 1\}$ ， $G(5)=\{0, 2, 3, \dots\}$ ， $f(5)=0$ ；
- $g(6)=\{\#(4), \#(1, 4), \#(2, 2)\}=\{2, 1, 0\}$ ， $G(6)=\{3, 4, \dots\}$ ， $f(6)=3$ ；
- $g(7)=\{\#(4), \#(1, 4), \#(2, 3)\}=\{2, 2, 0\}$ ， $G(7)=\{1, 3, 4, \dots\}$ ， $f(7)=1$ ；

👉 图 2 所示的局面 $S=(7, 3, 3)$ ，有 $\#S=f(7)+f(3)+f(3)=1+1+1=1$ ，故 S 胜。

👉 游戏 C 的初始局面 $S=(3, 4, 6)$ ，有 $\#S=1+2+3=01+10+11=0$ ，故 S 负。

七、结论

📖 此类博弈游戏的一般性解法：

- ☞ 用一个 n 元组 (a_1, a_2, \dots, a_n) ，来描述游戏过程中的一个局面。
- ☞ 用符号 $\#S$ ，表示局面 S 所对应的二进制数。
- ☞ 用符号 $\$(x)$ ，表示局面 (x) 的下一步所有可能出现的局面的集合。
- ☞ 定义集合 $g(x)$ ：设 $\$(x) = \{S_1, S_2, \dots, S_k\}$ ，则 $g(x) = \{\#S_1, \#S_2, \dots, \#S_k\}$ 。
- ☞ 令非负整数集为全集，集合 $G(x)$ 表示集合 $g(x)$ 的补集。
- ☞ 定义函数 $f(n)$ ： $f(n) = \min\{G(n)\}$ ，即 $f(n)$ 等于集合 $G(n)$ 中的最小数。
- ☞ 设局面 $S = (a_1, a_2, \dots, a_n)$ ， $\#S = f(a_1) + f(a_2) + \dots + f(a_n)$ ，采用二进制数的加法。

✉ 若 $\#S \neq 0$ ，则先行者有必胜策略；若 $\#S = 0$ ，则后行者有必胜策略。

📖 适用范围和限制条件：

- 甲乙两人取石子游戏及其类似的游戏；
- 每一步只能对某一堆石子进行操作；
- 每一步操作的限制，只与这堆石子的数目或一些常数有关；
- 操作在有限步内终止，并不会出现循环；
- 谁无法继续操作，谁就是输家。

📖 游戏 D (POI'2000, Stripes)：

📖 一排石子有 L 个，甲乙两人轮流从中取“紧紧挨着的” A 或 B 或 C 枚石子。

谁不能取了，谁就是输家。已知 A, B, C, L ，问甲乙二人谁有必胜策略。

😊 有了前面的结论，这个游戏就难不倒我们了。

八、总结

1. 从算法优化的角度

取石子游戏属于一类典型的博弈游戏。穷举所有的局面，理论上可以求得最优策略。但穷举的时空复杂度太高，本文所提出的解法，有效的控制了算法的时空复杂度，可以看作是对穷举法的一个优化。

优化算法的过程，可以看作是在优化局面的表示。首先，我们用一个 n 元组表示一个局面，这是很直观很容易想到的。因为我们只关心局面的胜负，于是得到了第一个性质：这个 n 元组是无序的。进一步分析发现， n 元组中如果出现两个相同的数字，则把它们消去，不影响局面的胜负。于是，我们改用集合来表示一个局面。最后，通过与二进制数的对比，又简化到用一个数来表示一个局面。

优化局面的表示，使得搜索量大大减少。那么，减少的搜索量都到哪里去了呢？举个例子，对于游戏 A 中的 5 个局面：(3, 3, 1), (1, 3, 3), (5, 5, 1), (2, 3):

- a. 采用 n 元组：这 5 个局面互不相同；
- b. 采用无序 n 元组：局面(3, 3, 1)和(1, 3, 3)相同；
- c. 采用集合：局面(3, 3, 1), (1, 3, 3), (5, 5, 1)都相同，可以用集合{1}表示；
- d. 采用二进制数：4 个局面所对应的二进制数都是 1，故都相同。

算法的优化，本质上是[避免穷举相同的局面](#)，即避免重复搜索。而优化的关键，就在于“相同局面”的定义。

“相同局面”的定义，必须能够反映游戏的性质。我们没有简单的按照局面的胜负，来对局面归类，就是这个原因。

2. 从算法构造的角度

人们认识事物的过程中，开始只是看到了各个事物的现象。这就是认识的感性阶段。在这个阶段中，还不能作出合乎逻辑的结论。随着研究的深入，这些感觉和印象的东西反复了多次，于是在人们的脑子里生起了一个认识过程中的突变，最后产生出合乎逻辑的结论。这就是认识的理性阶段。

人们认识事物的过程，就是由感性认识上升到理性认识的过程。具体到解这类游戏，就是要**从简单入手**。当我们遇到了一个复杂的问题，或许人人都知道从简单入手，但却并不是每个人都能从中得到一般性的规律。那么，我们究竟是如何**由浅入深**的呢？

两堆数目相等的石子——这是个很简单的局面。我们就由此入手，将一堆石子与一个子局面相类比，并得出了两个子局面相等时的结论。在此基础上，我们研究了局面的胜负和其子局面的关系，并得出结论：可以用集合来描述一个局面。但我们并没有停留在这一步，而是将局面的分解与二进制数的加法相类比，从而发现了局面与二进制数之间的关系。我们称这个过程为“**由此及彼**”。

通过分析“用集合来表示一个局面”的结论，我发现这实质上是简化了局面的表示，从而联想到能否进一步化简，比如说用一个数来表示。在解游戏 C 时，我们并不在意它与游戏 A 的规则有多大的区别，而是注意到它与游戏 A 有着相似的性质，从而想到用类似的方法解游戏 C。我们称这个过程为“**由表及里**”。

在解游戏 A 和 B 的过程中，我们积累了很多经验。但在解游戏 C 时，我们却仅仅提了解游戏 A 和 B 的精华：构造一个函数 f 。这就是“**去粗取精**”。

将局面与二进制数相类比，我们先试着把局面的胜负直接与二进制的 1 和 0 相类比。发现不妥后，再将其改为与二进制数来类比。这一步叫“**去伪存真**”。

“由此及彼、由表及里、去粗取精、去伪存真”，这就是由感性认识上升到理性认识的关键。