NIM“拈”游戏分析

问题

有 N 块石头和两个玩家 A 和 B，玩家 A 先将石头分成若干堆，然后按照 BABA……的顺序不断轮流取石头，能将剩下的石头一次取光的玩家获胜。每次取石头时，每个玩家只能从若干堆石头中任选一堆，取这一堆石头中任意数目（大于 1）个石头。

请问：玩家 A 有必胜策略吗？要怎么分配和取石头才能保证自己有把握取胜？

解法与分析

据说，该游戏起源于中国，英文名字叫做“NIM”，是由广东话“拈”（取物之意）音译而来，经由当年到美洲打工的华人流传出去，这个游戏一个常见的变种是将十二枚硬币分三列排成 [3，4，5] 再开始玩 。我们这里讨论的是一般意义上的“拈”游戏。

言归正传，在面试者咄咄逼人的目光下，你要如何着手解决这个问题？

在面试中，面试者考察的重点不是“what”——能否记住某道题目的解法，某件历史事件发生的确切年代，C++语言中关于类的继承的某个规则的分支等。面试者很想知道的是“how”——应聘者是如何思考和学习的。

所以，应聘者得展现自己的思路。解答这类问题应从最基本的特例开始分析。我们用 N 表示石头的堆数，M 表示总的石头数目。

当 N=1 时，即只有一堆石头——显然无论你放多少石头，你的对手都能一次全拿光，你不能这样摆。

当 N=2 时，即有两堆石头，最简单的情况是每堆石头中各有一个石子（1，1）——先让对手拿，无论怎样你都可以获胜。我们把这种在双方理性走法下，你一定能够赢的

局面叫作安全局面。

当 N = 2，M > 2 时，既然（1, 1）是安全局面，那么（1, X）都不是安全局面，因为对手只要经过一次转换，就能把（1, X）变成（1, 1），然后该你走，你就输了。既然（1, X）不安全，那么（2, 2）如何？经过分析，（2，2）是安全的，因为它不能一步变成（1，1）这样的安全局面。这样我们似乎可以推理（3, 3）、（4, 4），一直到（X, X）都是安全局面。

于是我们初步总结，如果石头的数目是偶数，就把它们分为两堆，每堆有同样多的数目。这样无论对手如何取，你只要保证你取之后是安全局面（X, X），你就能赢。

好，如果石头数目是奇数个呢？

当 M=3 的时候，有两种情况，（2, 1）、（1, 1, 1），这两种情况都会是先拿者赢。

当 M=5 的时候，和 M=3 类似。无论你怎么摆，都会是先拿者赢。

若 M=7 呢？情况多起来了，头有些晕了，好像也是先拿者赢。

我们在这里得到一个很重要的阶段性结论：

当摆放方法为（1, 1,…, 1）的时候，如果 1 的个数是奇数个，则先拿者赢；如果 1 的个数是偶数个，则先拿者必输。

当摆放方法为（1, 1,…, 1, X）（多个 1，加上一个大于 1 的 X）的时候，先拿者必赢。因为：

如果 1 有奇数个，先拿者可以从（X）这一堆中一次拿走 X−1 个，剩下偶数个 1——接下来动手的人必输。

如果有偶数个 1，加上一个 X，先拿者可以一次把 X 都拿光，剩下偶数个 1——接下来动手的人也必输。

当然，游戏是两个人玩的，还有其他的各种摆法，例如当 M = 9 的时候，我们可以摆为（2, 3, 4）、（1, 4, 4）、（1, 2, 6），等等，这么多堆石头，它们既互相独立，又互相牵制，那如何分析得出致胜策略呢？关键是找到在这一系列变化过程中有没有一个特性始终决定着输赢。这个时候，就得考验一下真功夫了，我们要想想大学一年级数理逻辑课上学的异或（XOR）运算。异或运算规则如下：

XOR（0, 0）= 0
XOR（1, 0）= 1
XOR（1, 1）= 0

首先我们看整个游戏过程，我们从N堆石头（M_1, M_2, …， M_n）开始，双方斗智斗勇，石头一直递减到全部为零（0, 0,…, 0）。

当 M 为偶数的时候，我们的取胜策略是把 M 分成相同的两份，这样就能取胜。

开始：（M_1, M_1）　　它们异或的结果是XOR（M_1, M_1）= 0

中途：（M_1, M_2）　　对手无论怎样从这堆石头中取， XOR（M_1, M_2）!= 0

我方：（M_2, M_2）　　我方还是把两堆变相等。XOR（M_2, M_2）= 0

…

最后：（M_2, M_2）　　我方取胜

类似的，若M为奇数，我们把石头分成（1, 1, …,1）奇数堆的时候，XOR（1, 1,…,1）$_{[奇数个]}$!=0。而这时候，对方可以取走一整堆， XOR（1, 1,…, 1）$_{[偶数个]}$=0， 如此下去，我方必输。

我们推广到 M 为奇数，但是每堆石头的数目不限于 1 的情况，看看 XOR 值的规律：

开始：（M_1, M_2, …, M_n）　　XOR（M_1, M_2, … M_n）=？

中途：（M_1’, M_2’, … M_n’）　　XOR（M_1’, M_2’, … M_n’）=？

最后：（0, 0, …, 0）　　XOR（0， 0， … 0）=0

不幸的是，可以看出，当有奇数个石头时，无论你如何分堆，XOR（M_1, M_2, … M_n）总是不等于 0！因为必然会有奇数堆有奇数个石头（二进制表示最低位为 1），异或的结果最低位肯定为 1。　[结论 1]

再不幸的是，还可以证明，当XOR（M_1, M_2, … M_n）!= 0 时，我们总是只需要改变一个M_i的值，就可以让XOR（M_1, M_2, …M_i’， … M_n）= 0。　[结论 2]

更不幸的是，又可以证明，当XOR（M_1, M_2, … M_n）= 0 时，对任何一个M值的改变（取走石头）， 都会让XOR（M_1, M_2, …M_i’， … M_n）! = 0。　[结论 3]

有了这三个“不幸”的结论，我们不得不承认，当 M 为奇数时，无论怎样分堆，总是先动手的人赢。

还不信？那我们试试看：当 M=9，随机分堆为（1，2，6）

开始：（1，2，6）

```
    1=0 0 1
    2=0 1 0
    6=1 1 0
  ---------
XOR=1 0 1        即 XOR（1，2，6）!=0
```

B 先手：（1, 2, 3），即从第三堆取走三个，得到（1，2，3）

```
    1=0 0 1
    2=0 1 0
    3=0 1 1
  ---------
XOR=0 0 0        所以，XOR（1，2，3）=0
```

A 方：（1, 2, 2）XOR（1, 2, 2）!=0。

B 方：（0, 2, 2）XOR （0, 2, 2）=0

……A 方继续顽抗……

B 方最后：（0, 0, 0），XOR（0, 0, 0）= 0

好了，通过以上的分析，我们不但知道了这类问题的答案，还知道了游戏的规律，以及如何才能赢。XOR，这个我们很早就学过的运算，在这里帮了大忙[1]。我们应该对XOR说Orz才对！

有兴趣的读者可以写一个程序，返回当输入为（$M_1, M_2, \cdots, M_n$）的时候，到底如何取石头，才能有赢的可能。比如，当输入为（3, 4, 5）的时候[2]，程序返回（1, 4, 5）——这样就转败为胜了！

扩展问题

1．如果规定相反，取光所有石头的人输，又该如何控制局面？

[1] 温馨提示：你还记得教我们XOR运算的老师么？这门课一定比较枯燥吧，如果当时能玩NIM这个游戏就好了。

[2] 提一句，这是一个不明智的分堆办法，不如分为（6, 6），这样必赢无疑。

2．如果每次可以挑选任意K堆，并从中任意取石头，又该如何找到必胜策略呢？