Working Backwards: Solution

We can represent the state of the game with three numbers (a, b, c), which means there are a, b, and c marbles of each color left. Say (a, b, c) is a winning position if perfect play starting from that position has a guaranteed win, and a losing position otherwise. (Order doesn't matter to whether the game state is a winning or losing position, so we might as well assume $a \leq b \leq c$ so that we don't have to write as many triples.)

A position is winning if (and only if) either you can win the game on that turn or else it is possible to make a move which leaves your friend with a losing position. Conversely, a position is losing if (and only if) no matter what move you make you leave your friend with a winning position.

The game states where you can win on your turn are those where all marbles have only one color left, i.e. when two of the numbers a, b, c are zero:

Win: (0,0,1), (0,0,2), (0,0,3).

The losing position with fewest marbles is $(0, 1, 1)$, since no matter which marble you take you leave your friend with the winning game position $(0, 0, 1)$; this means any state that can reach $(0, 1, 1)$ on the next turn is winning:

Win: (0,1,2), (0,1,3),
(1,1,1), (1,1,2), (1,1,3).

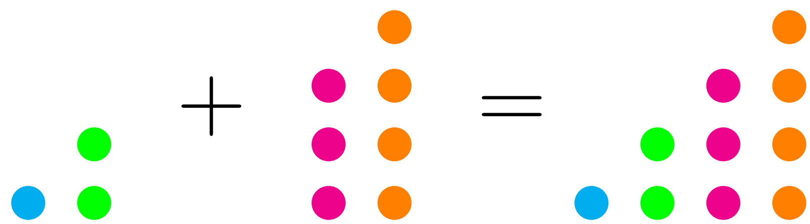
This means (0,2,2) must be a losing position, because it can only leave your friend with one of the winning positions (0,1,2) or (0,0,2), which also means

Win: (1,2,2), (0,2,3).

Finally, (1,2,3) is a losing position because it too can only leave your friend one of the winning positions (0,2,3), (1,1,3), (1,2,2), (1,1,2), (1,1,1), (0,1,2).

In conclusion, if you believe your friend is banking on this fact, or else can recognize and capitalize on it immediately, you should **decline** her offer.

This is the game of **Nim**. The game can starts with any number of "heaps," each heap having any number of items, and you take turns with your opponent removing any number of items from one heap per turn. Games can be added:



In game theory terms, the combined game is the **disjunctive sum** of the smaller games. The sum $A+B$ of two games A and B has players playing the games A and B "in parallel," meaning each turn a player performs a move in either game A or game B (until one of them is concluded, after which the players finish the remaining game), and the sum game is finished when both A and B are, with the final outcome the same as that of the final game.

Compare with simultaneous exhibitions where a high-ranking player plays multiple other players (in chess or Go, for example) at the same time. The disjunctive sum is like this, but played 1v1 instead of against multiple other players, and only the last-concluded game's outcome counts.

Maybe we can determine if the $A+B$ is a winning or losing position based on the positions of Nim games A and B? Consider the possibilities:

- $L+L=L$. If A and B are both losing positions, your opponent can play perfectly in both games and win both, hence the sum.
- $L+W=W$. If (say) A is a winning position and B is a losing position, then you can play a correct move in A to leave your opponent with two games in losing position, which we just said is a losing position.
- $W+W$ is indeterminate, as our previous work reveals - for example $(1)+(1)=(1,2)$ is a losing position but $(1)+(2)=(1,2)$ is winning.

Perhaps we can study some special cases and generalize? The simple game $(1,1,\cdots,1)$ with n one-item heaps is winning or losing based on n's parity, i.e. it is a winning position for odd n and a losing position for even n.

If we analyze the two-heap game (a, b) for small values of a and b we will find it is only a losing position when $a = b$. What makes equal-size pairs of heaps so special? For the game (a, a), you cannot take all of one heap because then your opponent can win next turn. But then whatever number of items you take from one heap, your opponent can mirror your move by taking an equal number of items from the other heap! This generalizes: given any game A, the game $A + A$ must be a losing position, since whatever move you do in one of the A games, your opponent can copy it in the other A game.

If we analyze the three-heap game (a, b, c) for small values of a, b, c, no obvious pattern emerges. (We will find out in a bit that the three-heap games are the key to solving the full game with any number of heaps.)

Consider the effect of adding one heap to a game. For any game A, there is *at most* one number a for which $A + (a)$ is in a losing position. Indeed, if there were two such numbers $a < b$, then faced with $A + (b)$ you could remove items from (b) to leave your opponent with the losing position $A + (a)$, which means $A + (b)$ was a winning position for you, a contradiction. The value a (which will turn out to always exist) is called the **nim sum** of A.

To anyone familiar with computer science, this may feel like a **parity bit**, or more generally a **checksum**. Computers deal in bits, or `1`s and `0`s (corresponding to the presence or absence of electrical signals in circuits, or to magnetic poles in physical media, or to `True` and `False` respectively in general). Computers often append a parity bit to the end of a code (string of bits) which is the `XOR` (exclusive or) of previous bits. In practice this means the parity bit has the same parity as the number of `1`s in the code (**parity** means whether a number is even or odd).

A checksum is a larger block of data that functions the same way. Parity bits and checksums are examples of error-detection systems. More sophisticated coding schemes can be used to not only detect but also correct reasonable levels of errors in signals travelling over noisy channels on-the-fly. Modern computers and the internet as we know it wouldn't be possible without error correction fixing flipped bits caused by interference like stray particles.

Just as a parity bit is the unique bit added to the end of a code to make the total `XOR` equal `0`, the nim-sum of a game is the unique heap size (possibly zero) to add to make it a losing position, hence the analogy.

Let's use the notation Σ_A for the nim sum of a game A. By definition, $A + (\Sigma_A)$ and $B + (\Sigma_B)$ are losing positions, as is $A + B + (\Sigma_{A+B})$. That means the sum of all three of them is also a losing position:

$$A + (\Sigma_A) + B + (\Sigma_B) + A + B + (\Sigma_{A+B}).$$

We said earlier $L + L = L$ and $L + W = W$, which means subtracting any losing game (if possible) from a sum does not change winning or losing position. As $A+B+A+B$ is a losing position, we can subtract it to conclude

$$(\Sigma_A) + (\Sigma_B) + (\Sigma_{A+B})$$

is also a losing position, which means not only is Σ_{A+B} the nim-sum of A and B, it is also the nim-sum of $(\Sigma_A) + (\Sigma_B)$! This means we can replace the last two heaps of a game $(a_1, \cdots, a_{n-1}, a_n)$ with their nim-sum as a single heap and the overall nim-sum is unaffected. Thus, finding the nim-sum of any game is reduced to finding the nim-sum of two-heap games!

Let's pick notation that indicates we're considering a binary operation: let $a \oplus b$ denote the nim sum of $(a) + (b)$. We've found $1 \oplus 2 = 3$ in this problem, for instance. It takes quite a bit of legwork (working out case after case by hand) before we can observe any patterns in a nim-sum table:

0	1	2	3	4	5	6	7
1	0	3	2	5	4	7	6
2	3	0	1	6	7	4	5
3	2	1	0	7	6	5	4
4	5	6	7	0	1	2	3
5	4	7	6	1	0	3	2
6	7	4	5	2	3	0	1
7	6	5	4	3	2	1	0

Every entry in the table is the nim-sum of its row's first number and its column's top number.

The diagonal is filled with 0s (as we said earlier, $A + A$ is always a losing position, which implies $a \oplus a = 0$). Also notice there are many runs of consecutive numbers within rows and columns of different sizes.

There also seems to be interlacing and reversing effects - for example, in the row beginning with 1, the numbers after 0 are just the same numbers above them but every consecutive pair is swapped. The effect is more pronounced if we subdivide the 8×8 array into a 2×2 array of 4×4 arrays: each 4×4 array is identical to the catercorner 4×4 array. The same is true if we subdivide the 4×4 arrays into 2×2 arrays, and this pattern continues if we zoom out to look at 8×8 arrays within a 16×16 or beyond.

This fractal-like structure (self-similarity at different scales) is reminiscent of numbers represented in **binary**. In our decimal (i.e. base-10) number system (chosen because we have ten fingers), each digit represents higher powers of ten, so for instance $100 = 10^2$ and $1000 = 10^3$. In binary, the digits represent powers of two, so `111` in binary is $4 + 2 + 1 = 7$ in decimal for example.

binary	decimal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

000	001	010	011	100	101	110	111
001	000	011	010	101	100	111	110
010	011	000	001	110	111	100	101
011	010	001	000	111	110	101	100
100	101	110	111	000	001	010	011
101	100	111	110	001	000	011	010
110	111	100	101	010	011	000	001
111	110	101	100	011	010	001	000

In the left table of binary representations, notice the units digit alternates between `0` and `1`, the twos digit alternates between two `0`s and two `1`s, the fours digit alternates between four `0`s and four `1`s, and this pattern continues.

On a hunch, let's rewrite the nim-sum table in binary. It may not be obvious, but this is also the table for bitwise `XOR`! That is, a binary digit of $a \oplus b$ is `0` if the corresponding binary digits of a and b match, and is `1` otherwise.

This completely solves the game of Nim. For example, consider the game $(1, 2, 3, 4)$. The nim-sum is the bitwise `XOR` of the numbers $1, 2, 3, 4$:

$$
\begin{array}{rr}
 & \texttt{0001} \\
 & \texttt{0010} \\
 & \texttt{0011} \\
\oplus & \texttt{0100} \\
\hline
 & \texttt{0100}
\end{array}
$$

The nim-sum is $1 \oplus 2 \oplus 3 \oplus 4 = 4$. Since the nim-sum is not zero, we conclude $(1, 2, 3, 4)$ is a winning position. In fact, this gives us the strategy for perfect play: always leave your opponent with a nim-sum of zero for the start of their turn.

The earliest European references to Nim are from the 1500s, but it's possible the game originated in China, and some of its many variations have been played since ancient times. The first published mathematical solution was in 1901, for comparison, so a lot of time went by before the solution was finally found. Don't get hung up if you didn't see it; almost nobody does!