Nim

Jason Zhang

Brief Motivation

What if I told you there was a simple game that underlies a bunch of competitive programming problems?

That would be quite convenient, and luckily for us there is such a game, which is the subject of this lecture.

Disclaimer

This lecture is for beginners, and the goal is to try to build intuition and explain why things are the way they are.

Do note that even if you think you're familiar with the subject of Nim, it might still be worthwhile to pay attention! This lecture will cover some of ideas on how to think about solving problems in general, as well as an in-depth explanation on why the solution to Nim is the way it is.

Nim

Rules

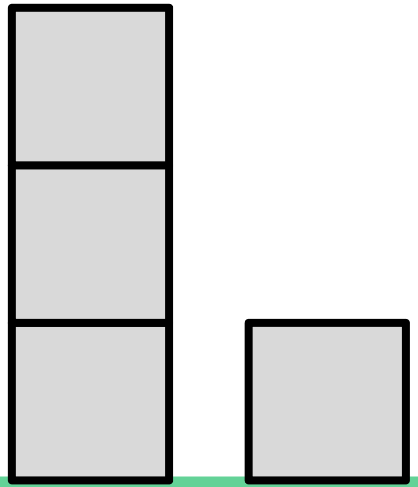
Imagine there you are playing a game with some number of piles of blocks. The piles may have varying heights. Alice (Player 1) and Bob (Player 2) take turns, each turn allows them to remove any number of blocks from a single pile (including all of them!). Blocks that are removed are no longer considered. The person who takes the last block wins!

The question is, given the number of piles and how tall each pile is, can we figure out who wins if both players play perfectly?

Example

Imagine that there are 2 piles as shown below.

We will run through an example game with Alice and Bob.

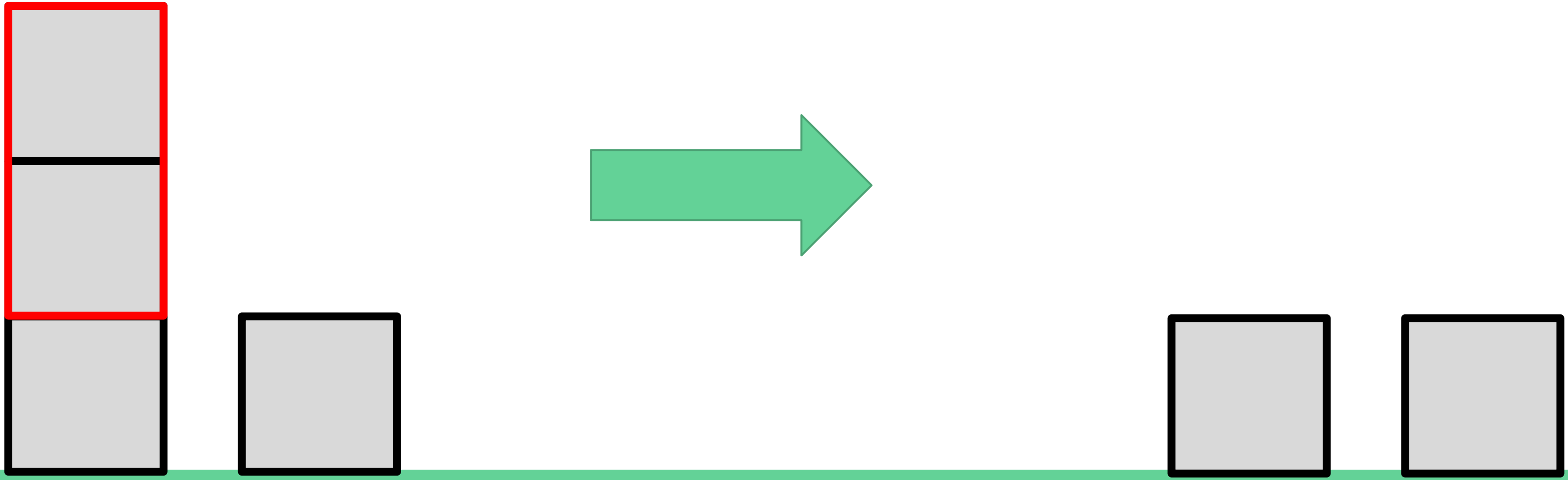


Turn 1

Alice decides to take 2 blocks from pile 1.

Before (with move highlighted)

After

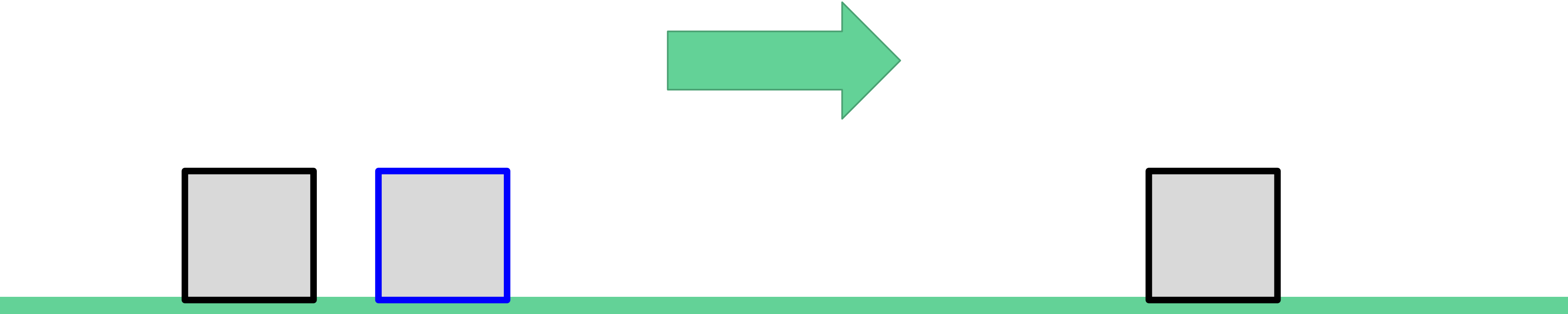


Turn 2

Bob takes 1 block from pile 2.

Before (with move highlighted)

After



Turn 3

Alice takes the **last** block. Alice wins!

Before (with move highlighted)

After (Alice wins!)

Towards a General Solution

Simulating it out is definitely NOT going to work. Imagine I gave you 10^6 piles, each of heights varying anywhere between 1 and 10^9, and I asked you who wins. It will be quite a long time before you solve just that one scenario!

Obviously, there must be some trick we can use to solve Nim without looking at every single possible move combination.

Simpler Cases

Analyzing Simple Cases

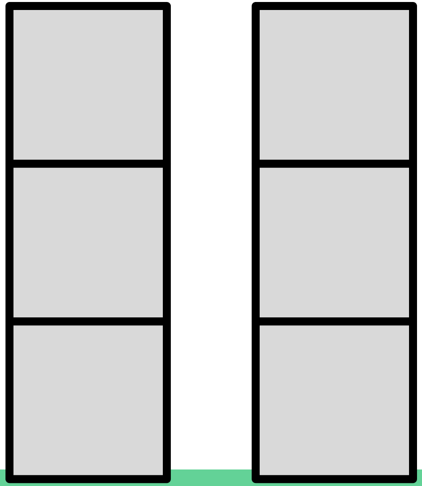
Always look at the easy cases to gain intuition!

If there is only 1 pile, then Alice always wins (she instantly grabs the whole pile).

If there are 2 piles, things get a little more interesting.

Who wins?

Take a minute to try to figure out who wins in this game of Nim.



Answer: Bob

“But why?”

This is a great question! There are many different ways to think about this (e.g., induction on winning/losing states), but I will try to explain it in a beginner friendly way :)

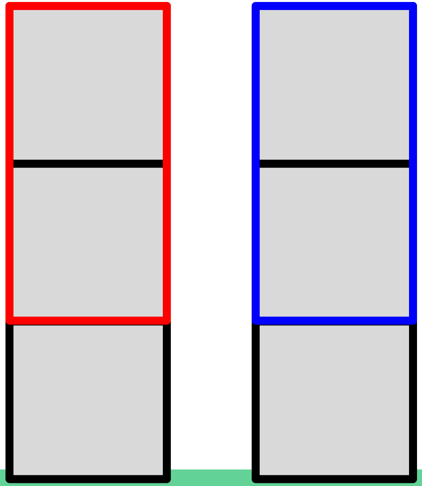
Let us simplify the rules of the game a bit:

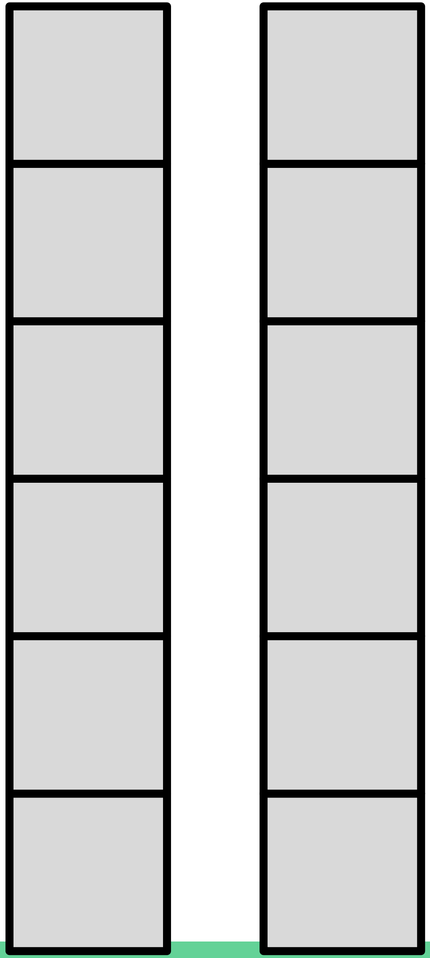
Alice can only take from the first pile, Bob only from the second.

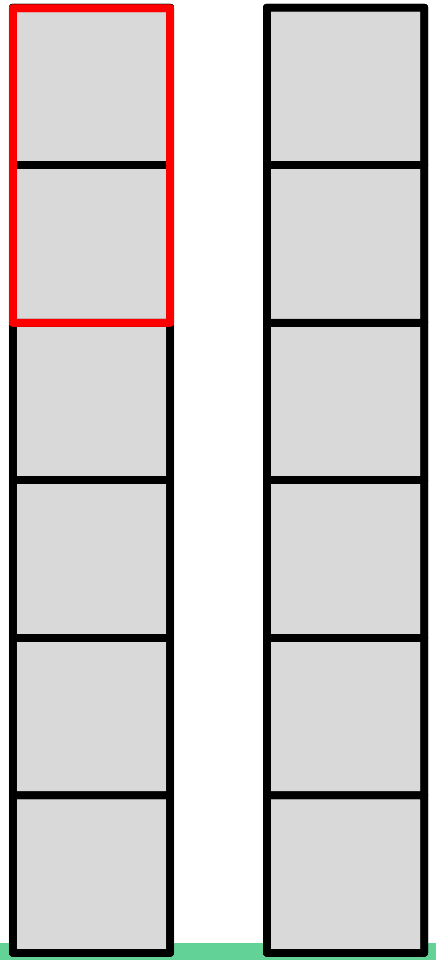
Copying

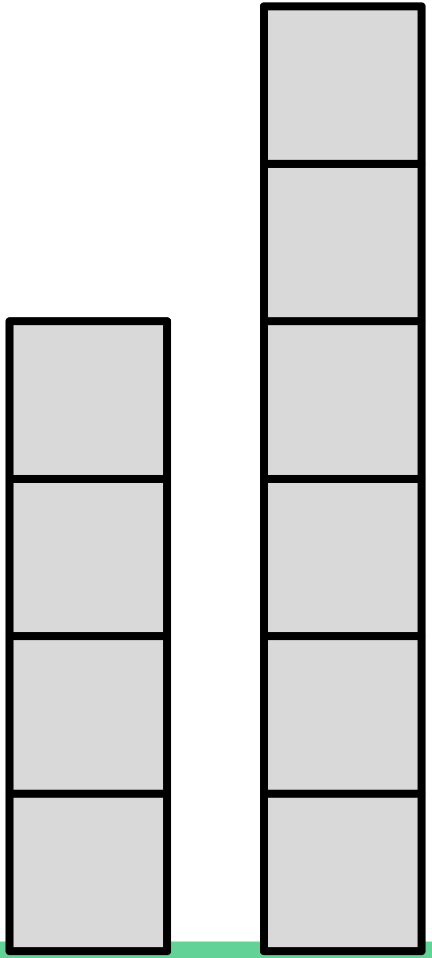
Now, imagine Bob just copies whatever Alice does on the first pile.

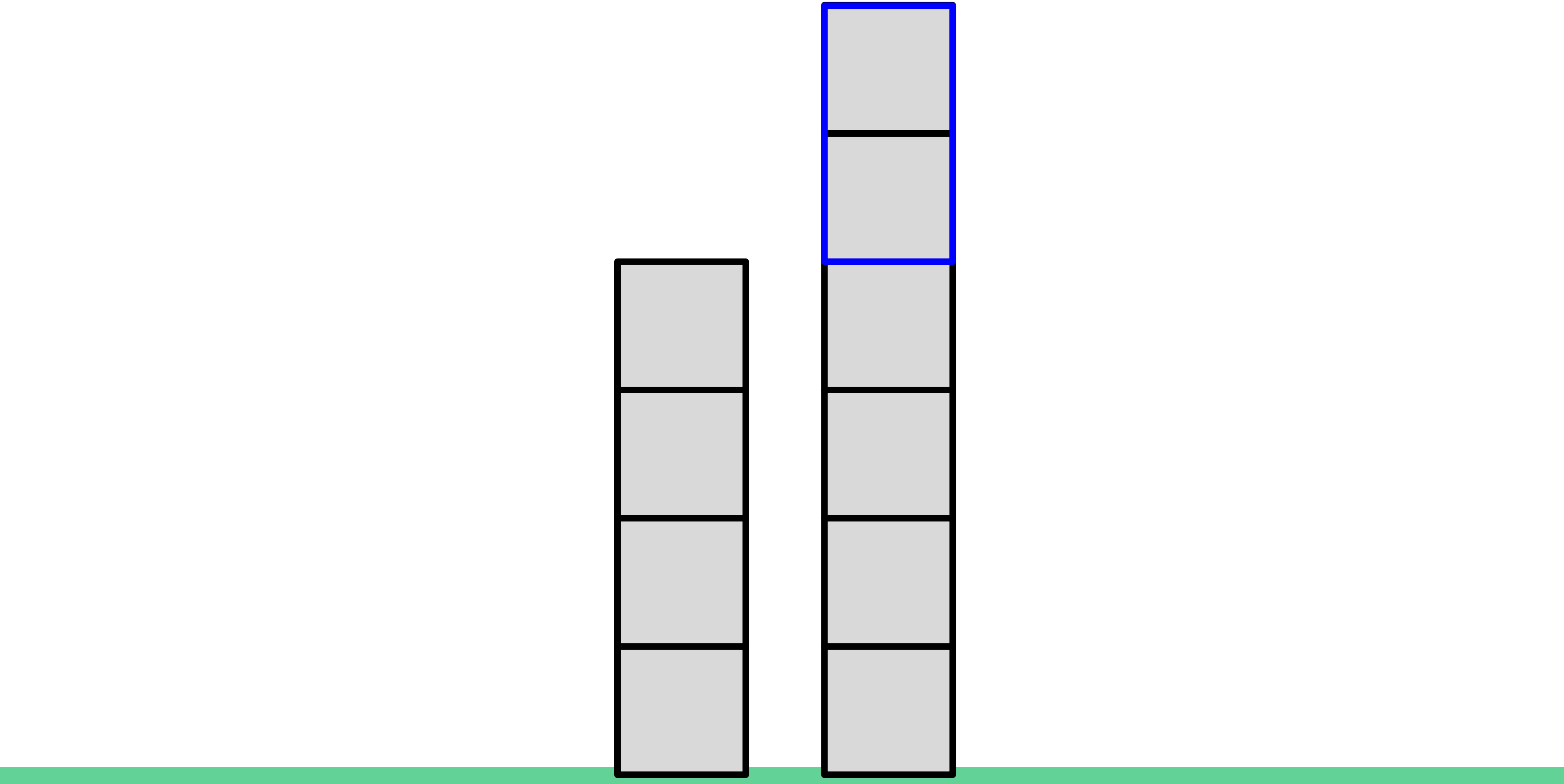
Eventually, Alice will have to make her pile empty. Bob will copy, making his pile empty as well, but he does this **after** Alice does, meaning he took the last piece and thus he would win. This works no matter how tall the piles are, just as long as they are of the same height.

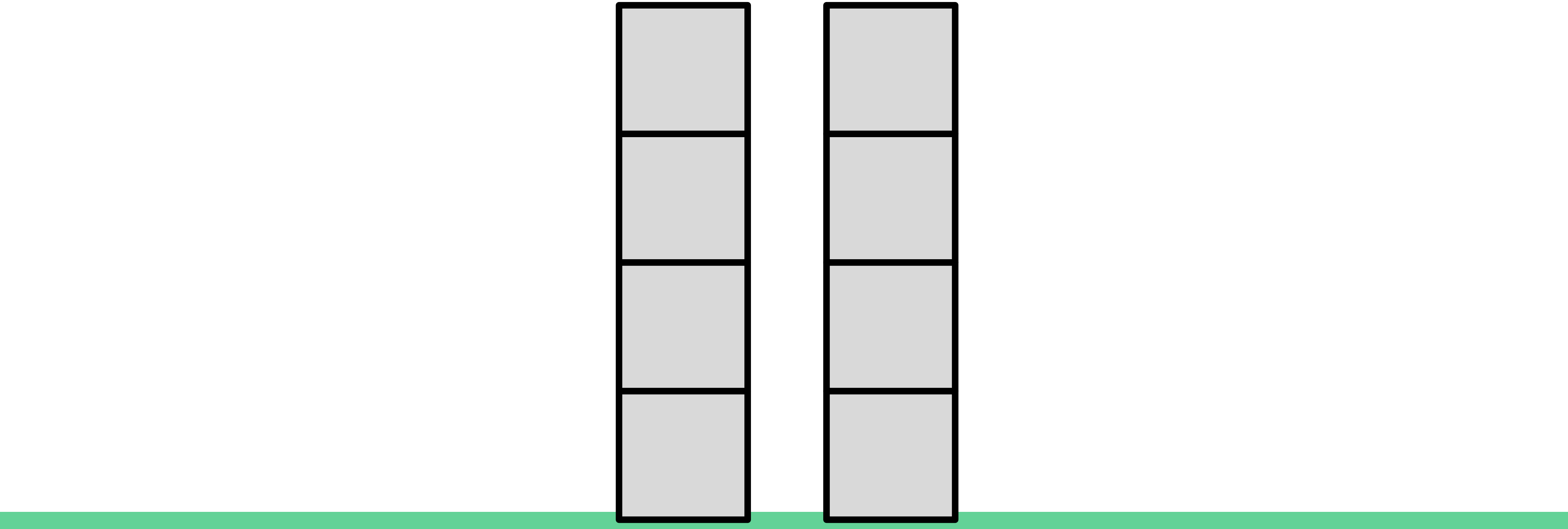


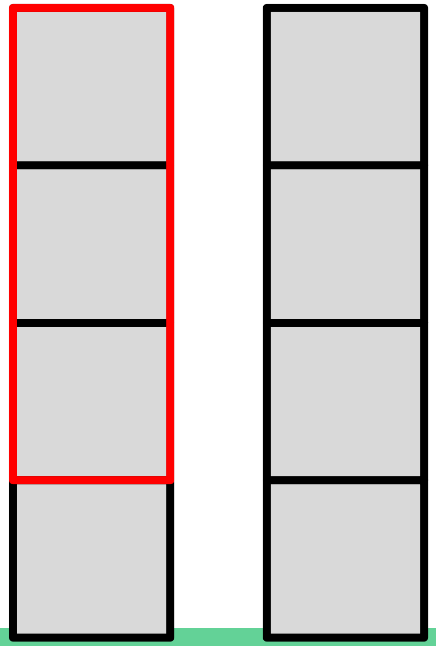


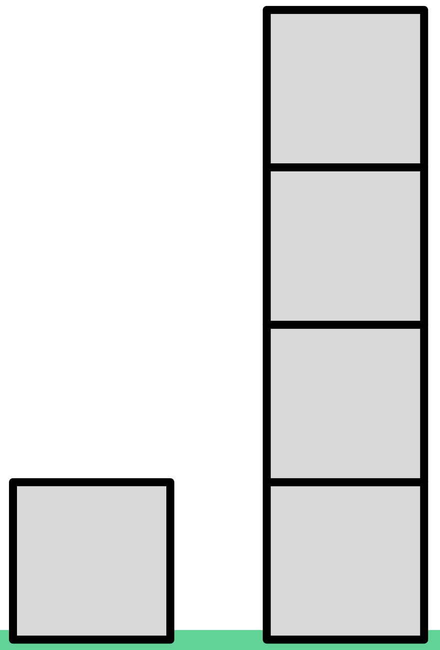


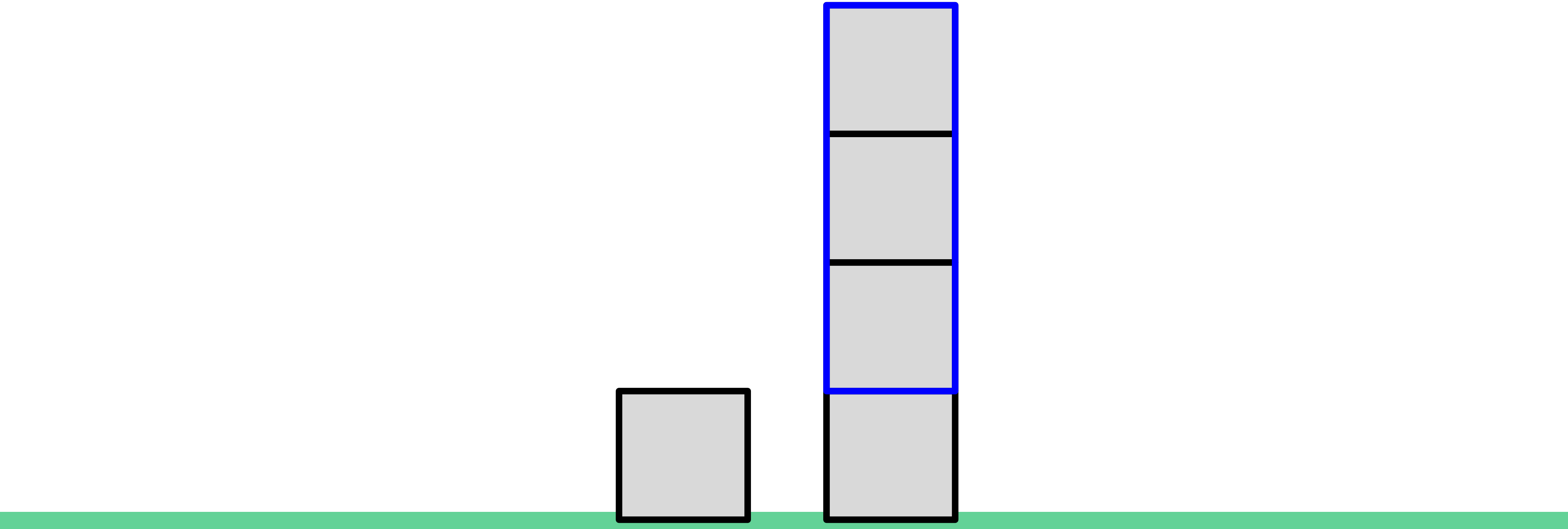


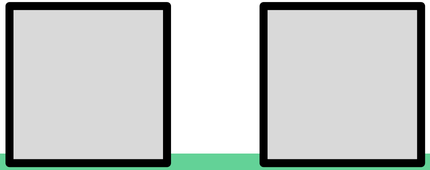


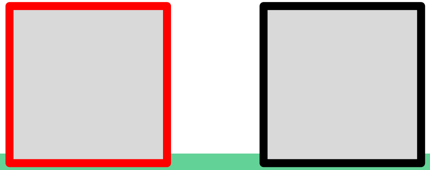


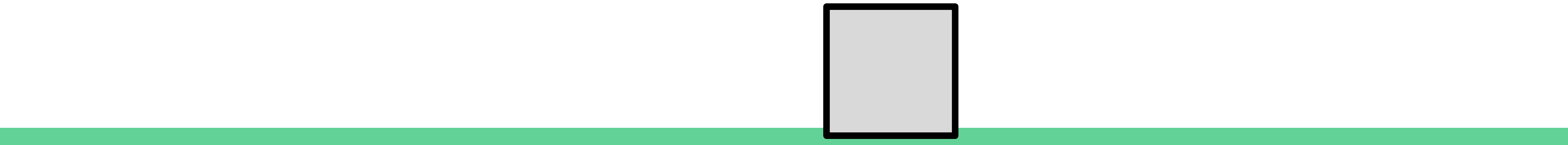


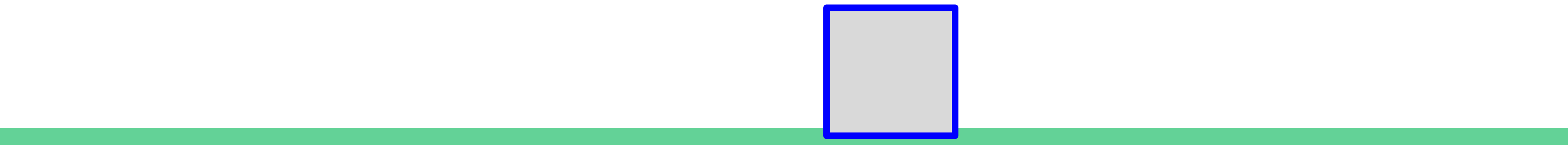












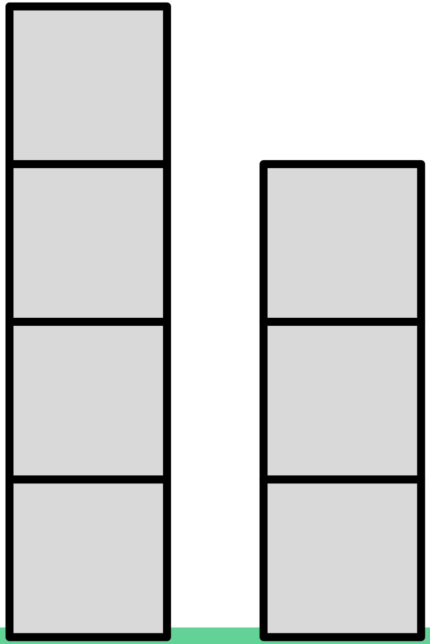
Bob wins! :)

What this shows:

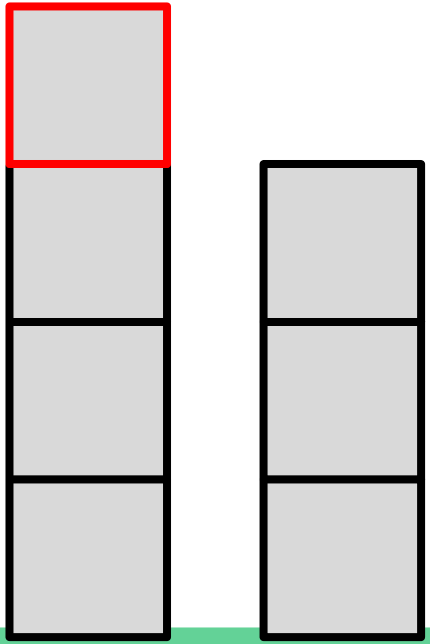
Hopefully, this convinces you that if there are two piles of equal height, then the person who goes second wins.

But what if they are not of equal heights?

Visual Explanation

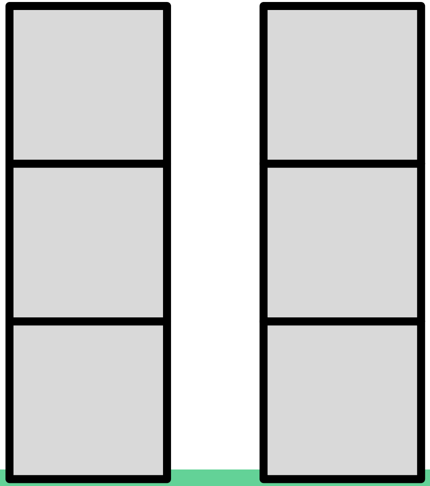


Visual Explanation



Visual Explanation

(Bob's turn)



Explanation

In case you didn’t catch it, Alice made the piles of equal height again! But then, it is Bob’s turn to go first, with Alice going second. And since we said whoever goes second, wins, in that case, Alice wins!

A Quick Note

Remember that we restricted Alice to the first pile and Bob to the second.

The key is, that restriction doesn't matter. Even if it didn't exist, everything we just said would still hold true!

If Alice suddenly decided to take from the second pile, Bob would respond in the first pile.

In other words, no matter what, two piles of equal height is a winning position for the second person to go.

The General Strategy

Idea

Is there a way we can generalize this idea of “equal heights” and “unequal heights” of two piles to n piles?

No, just checking if they are all equal heights no longer works.

We will be forced to dive into the mathematics to truly understand this.

Sketch

Our idea will fundamentally try to mirror the equal/unequal heights of the two piles case.

Essentially, we will argue that there is some general property that behaves exactly like the equal/unequal heights. This property dictates who wins the game.

Also, applying this property to just two piles changes it back to equal/unequal heights.

Binary

Did you know? We can express all of the positive integers as sums of powers of two!

For example: $13 = 2^3 + 2^2 + 2^0 = 8 + 4 + 1 = 13$ ✅

For each power of two, we will indicate whether we include it in our sum or not. The rightmost power of two represents 2^0.

This is how binary works!

$$13 = 1101_2$$

2^3 2^2 2^1 2^0

Balanced vs. Unbalanced

Now we introduce the property we are looking for.

Imagine you wrote down all of the heights of the piles in binary, and aligned them according to the rightmost digit.

Count the number of 1s in each column. If every column has an even number of 1s, then a position is **balanced**. Otherwise, it is unbalanced.

Example

Imagine we had towers of heights 2, 3, 6, and 7.

Binary:

010
011
110
111

Number of 1s: 2 | Number of 1s: 4 | Number of 1s: 2

Therefore, this game is **balanced**.

Our Argument

We argue that being **balanced** is the general version of the two piles having **equal** heights. Therefore, the second player in a **balanced** position wins, while the first player wins in an **unbalanced** position.

Unfortunately, there is no good explanation for why this works (the most mathematically accurate explanation involves a lot more knowledge needed).

We will attempt to develop an explanation similar to the equal/unequal piles.

Important Observations

1. When there is nothing left, it is a balanced position, as there are no 1’s.
2. Whenever someone takes anything from a balanced position, it becomes unbalanced.
3. One can always make an unbalanced position, balanced.

Notice how this mirrors the equal/unequal piles:

1. When there is nothing left, the two piles are of equal height (both 0).
2. Whenever someone takes anything from the two equal piles, they will become unequal afterwords.
3. One can always make unequal towers, equal.

Proof.

1. Obvious.
2. Think about the binary representation of the pile you are about to take away from. Let's say it is 11000101. When you take any number away, this representation will change. Now, all the piles were in perfect, even harmony beforehand. If you change this number in anyway, at least one of the columns will get an extra 1 or miss a 1. An even number + 1 or - 1 is odd, which ruins the balance.
3. Take the pile with the largest unbalanced of two, 2^k. Notice that $2^k - 1 = 11...11$ and so you can "manipulate" that binary string however you please. And since an odd number + 1 or - 1 will be even, you can balance the piles out.

Proof (pt. 2).

And thus, to conclude, we can combine all these observations.

Notice that one can force the game to be:

… ➔ Balanced ➔ Unbalanced ➔ Balanced ➔ …

And remember, taking the final blocks would **balance** the position, meaning that whoever receives a balanced position is doomed to lose. And whoever plays on an unbalanced position would win.

Thus, we can decide Nim. :)

Example

Consider the towers of 2, 3, 6, 7, again. This position is **balanced**. An equivalent way to say that is that

$$2 \oplus 3 \oplus 6 \oplus 7 = 0.$$

If you don't know what $\oplus$ (XOR) means, feel free to search it up after the lecture.

By applying our decision algorithm, we can realize that **Bob wins**.

Generalizations, Briefly

Why?

The point of Nim is that is generalizes to a lot of different games! That is, a bunch of different games are secretly Nim in disguise!

Here are some topics you might want to research about:

1. Grundy Numbers
2. Sprague-Grundy Theorem
3. Transfinite Nim

Simple Challenge

There is a variant called Misère Nim, where the person NOT to take the last block wins.

How can we decide that in general?

Further Reading

Here are some specific readings you might wanna do if you want to know more!

1. https://cp-algorithms.com/game_theory/sprague-grundy-nim.html. Introduction to Sprague-Grundy.
2. https://codeforces.com/blog/entry/66040. A more formal perspective on Nim and Sprague-Grundy Theorem.
3. https://codeforces.com/blog/entry/103785. Another good formal blog on Sprague-Grundy.
4. https://codeforces.com/blog/entry/85984. Transfinite generalization of Nim.
5. https://jdh.hamkins.org/transfinite-nim/. Another more technical post on Transfinite Nim.

Thank you

:)