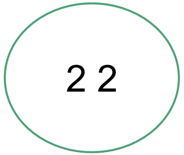
Nim Games

250H

How to Play

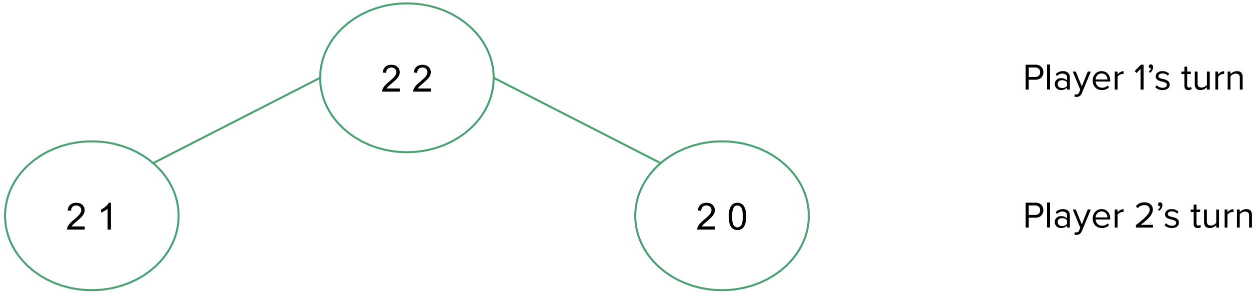
- To players take turns removing objects from distinct piles
  - You can have any number of piles and any amount of objects in each pile
- Each player must remove at least 1 object and may remove any number of objects as long as they all come from the same pile
- Depending on the version: the goal of the game is either to
  - Avoid taking the last object
  - To take the last object

We can use game trees to look at all possible games (the players are playing perfectly here)

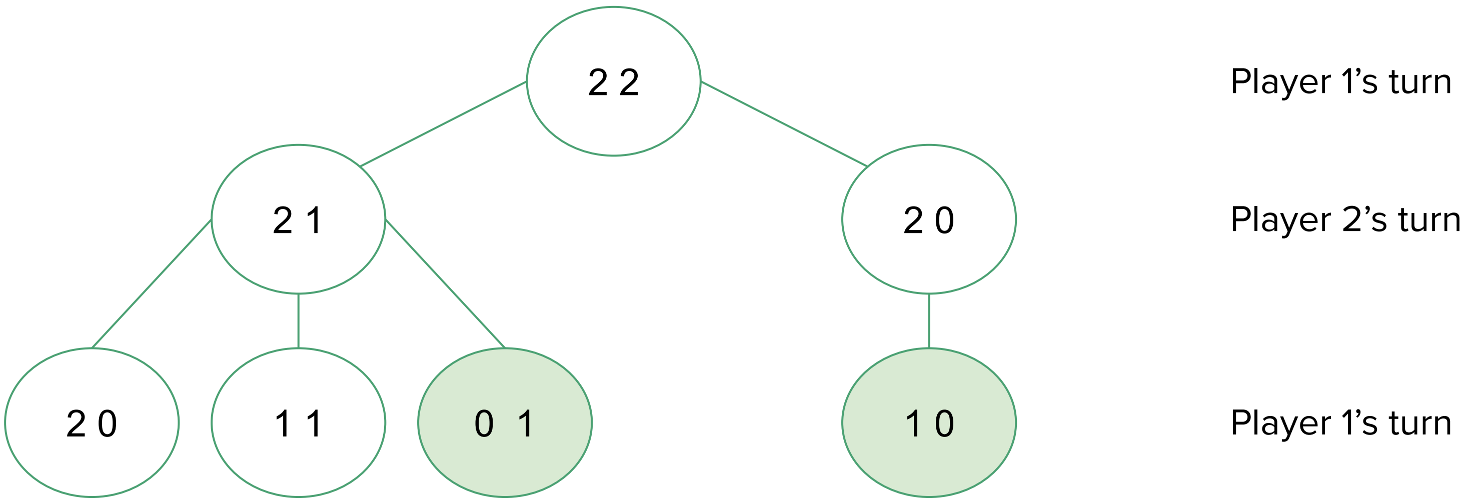


Player 1’s turn

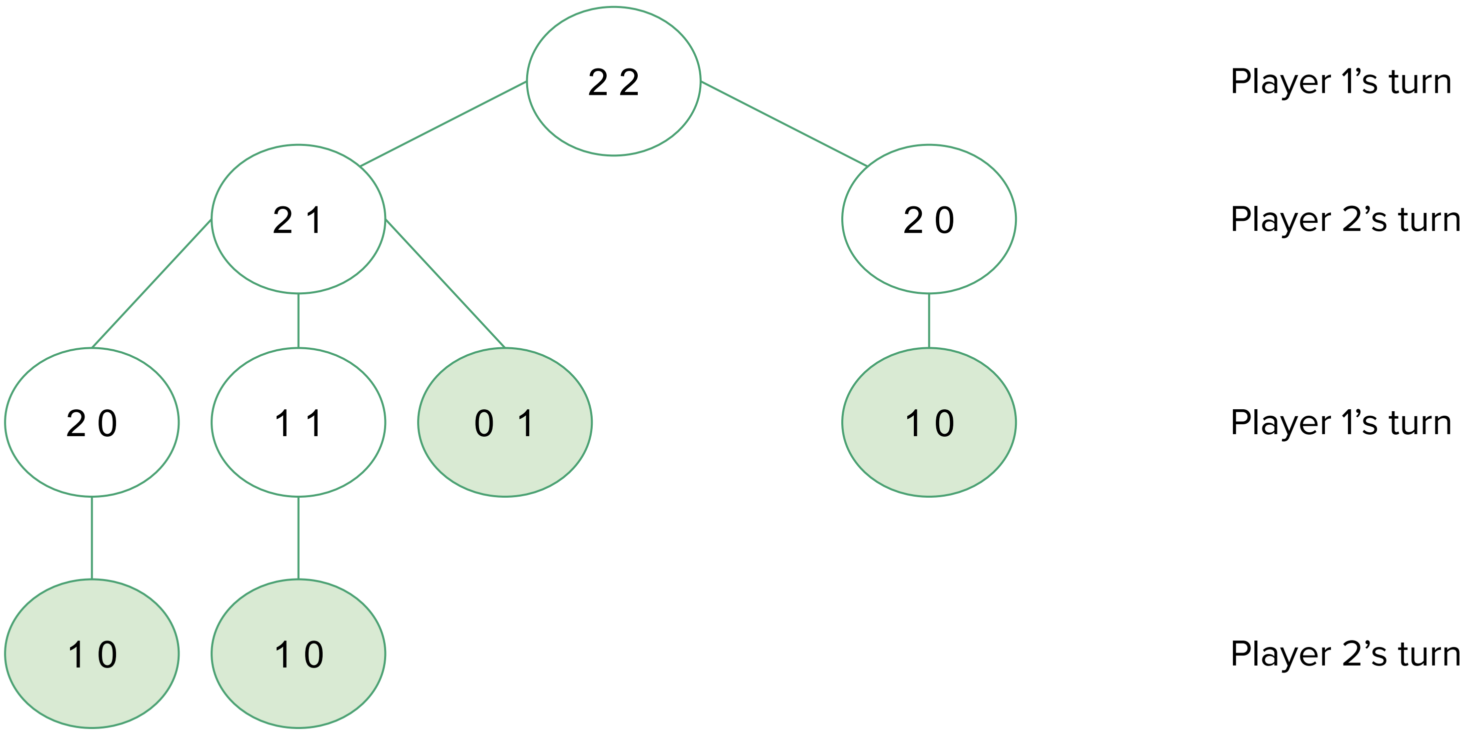
We can use game trees to look at all possible games (the players are playing perfectly here)



We can use game trees to look at all possible games (the players are playing perfectly here)



We can use game trees to look at all possible games (the players are playing perfectly here)



Consider a 2 pile game of Nim where you win if you pick up the last stone. Prove if both piles of stones have n stones each and it's the first player's turn, the second player can always win.

Base Case: If both piles have 0 stones in them, the first player loses

Inductive Hypothesis: Assume that for some $n \geq 0$ and $0 \leq i < n$. If both piles have i number of stones and it's the first player's turn, the second player can win.

Inductive Step: Consider a game of Nim in which there are two piles of stones, A and B, with n stones in each. Without loss of generality, let A be the pile that the first player chooses to remove stones from.

The first player must remove k stones from pile A such that $1 \leq k \leq n$. So, we have $n - k$ stones in pile A and n stones in pile B.

If the second player removes k stones from pile B, both piles have $n - k$ stones in each.

By the induction hypothesis, the second player can now win this game because there are two piles with $n - k$ stones in each.

What is the winning strategy?

- Need to write the sizes of the piles in binary
- Add those numbers up but not in the usual way (AKA use XOR)
    - If the number of 1's in a column is odd, you write a 1 underneath it
    - If it's even, you write a 0 underneath it.
    - Doing this for each column gives a new binary number, and that's the result of the Nim addition.

What is the winning strategy?

- Need to write the sizes of the piles in binary
- Add those numbers up but not in the usual way (AKA use XOR)
  - If the number of 1’s in a column is odd, you write a 1 underneath it
  - If it's even, you write a 0 underneath it.
  - Doing this for each column gives a new binary number, and that's the result of the Nim addition.
- Example:
  - Pile 1 has 2 objects
  - Pile 2 has 3 objects
  - $$\begin{array}{r} 10 \\ +\,11 \\ \hline 01 \end{array}$$

What is the winning strategy?

- Charles Bouton studied this game and figured out two things
  - Suppose it's your turn and the Nim sum of the number of objects in the pile is equal to 0
    - The Nim sum of the number of objects after your move will not be equal to 0
  - Suppose it's your turn and the Nim sum of the number of objects in the pile is not equal to 0
    - Then there is a move which ensures that the Nim sum of the number of objects in the pile after your move is equal to 0

What is the winning strategy?

- Let player 1 go first and the Nim sum of the number of objects in the piles not be equal to 0
- Player 1’s strategy: if possible always make a move that reduces the Nim sum after your move to 0
- This would then mean that whatever player 2 does next, the move would turn the next Nim sum into a number that's not 0
- Player 1 wins IFF there is a move he can make that puts the game into a Player 2 win position

Variant: You have 1 pile. Players can only remove a square number of objects. The player who removes the last object wins

- What is the winning strategy?
  - Let 0 be bad and 1 be good
  - If all numbers 1 .. N have been labeled as either bad or good, then
    - The number N+1 is bad if only good numbers can be reached by subtracting a positive square
    - The number N+1 is good if at least one bad number can be reached by subtracting a positive square
  - The winning strategy of the game: Try to pass on a bad number to your opponent

Variant: You have 1 pile. Players can only remove 1, 2, or 3 objects. The player who removes the last object wins

- What is the winning strategy?
  - If there are only 1, 2, or 3 objects left on your turn, you take all of them
  - If you have to move when there are 4 objects you will always lose
    - No matter how many you take, you will leave 1, 2, or 3
  - If there are 5, 6, or 7 objects, you can win by taking just enough to leave 4 objects
  - The winning strategy of the game: At the end of your turn, make it so that your opponent is taking from a multiple of 4 objects

Variant: You have 1 pile. Players can only remove 1, 3, or 4 objects. The player who removes the last object wins

- What is the winning strategy?
  - If there are only 1, 3, or 4 objects left on your turn, you take all of them
  - If you have to move when there are 2 objects you will always lose
    - You will leave 1
  - If there are 5, you can win by taking 3 objects
  - If there are 6, you can win by taking 4 objects
  - If you have to move when there are 7 objects you will always lose
  - The winning strategy of the game: At the end of your turn, make it so that your opponent is taking from a pile that is equivalent to 2 or 0 mod 7

Variant: You have 2 piles. Players can remove as many as they want from either OR the SAME amount from both. A player wins when they remove the last object.

- What is the winning strategy?
  - Any position in the game can be described by a pair of integers (n, m) with $n \leq m$, where n and m are the piles
  - The strategy of the game revolves around cold positions and hot positions:
    - Cold Position: the player whose turn it is to move will lose when playing perfectly
    - Hot Position: the player whose turn it is to move will win when playing perfectly
    - The optimal strategy from a hot position is to move to any cold position
  - The classification of positions into hot and cold can be looked at recursively:
    - (0,0) is a cold position
    - Any position from which a cold position can be reached in a single move is a hot position
    - If every move leads to a hot position, then a position is cold.