# Subtraction

- Turn-based game
- 2 players ⇝ First, Second
- Setup : N tokens to begin with
- Rules : a player removes 1, 2, or 3 tokens on their turn
- Winning condition : lose if stuck

(N is a non-negative integer)

N tokens

moves → removing 1, 2, or 3 tokens (if available ;

ends ⇝ 0 tokens left  no negative tokens allowed )

the last player to move wins.

Examples of who wins*

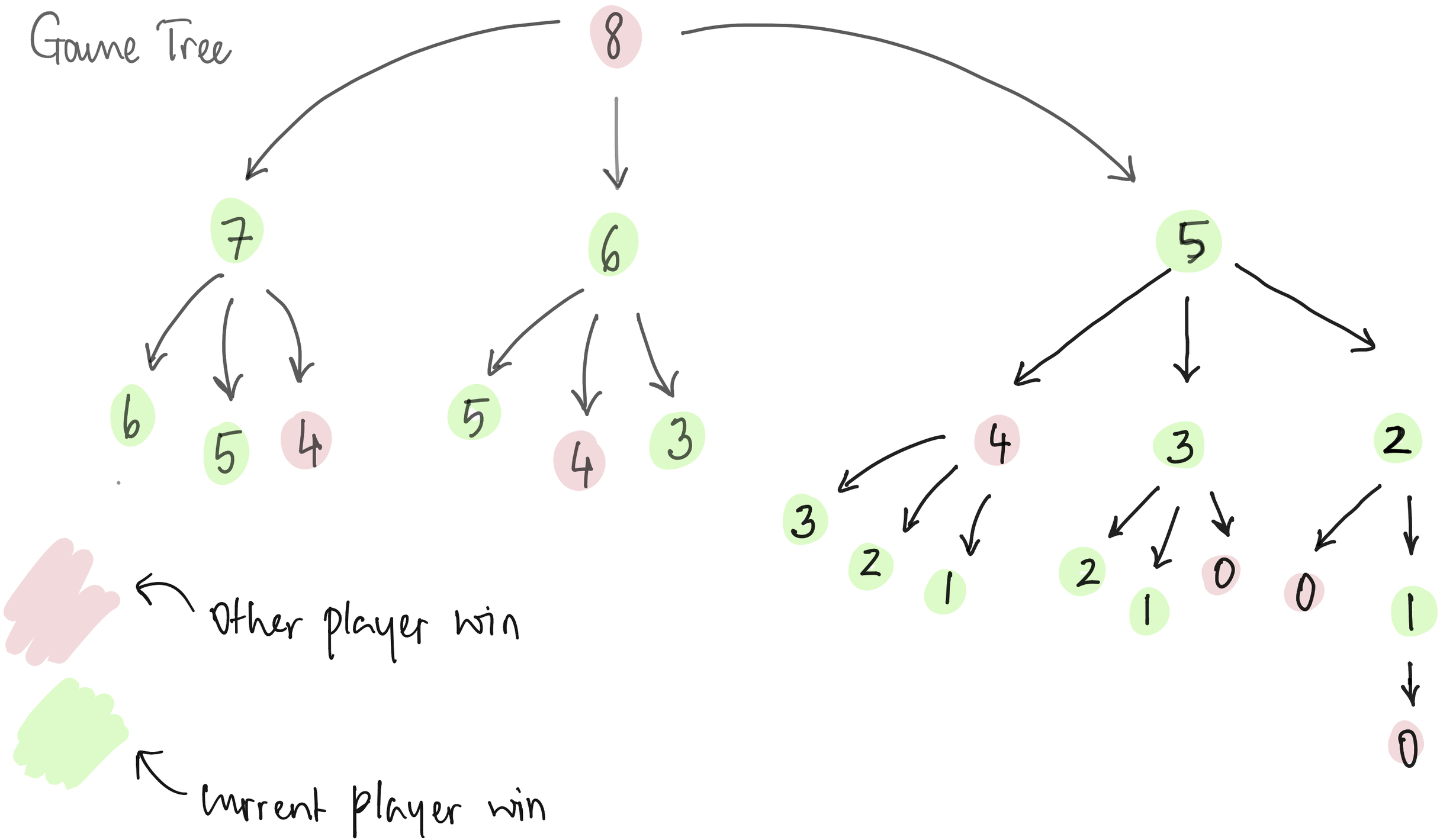
→ N = 1, 2, or 3 → first player win

→ N = 4 → second player win

→ N = 5, 6, or 7 → first player win

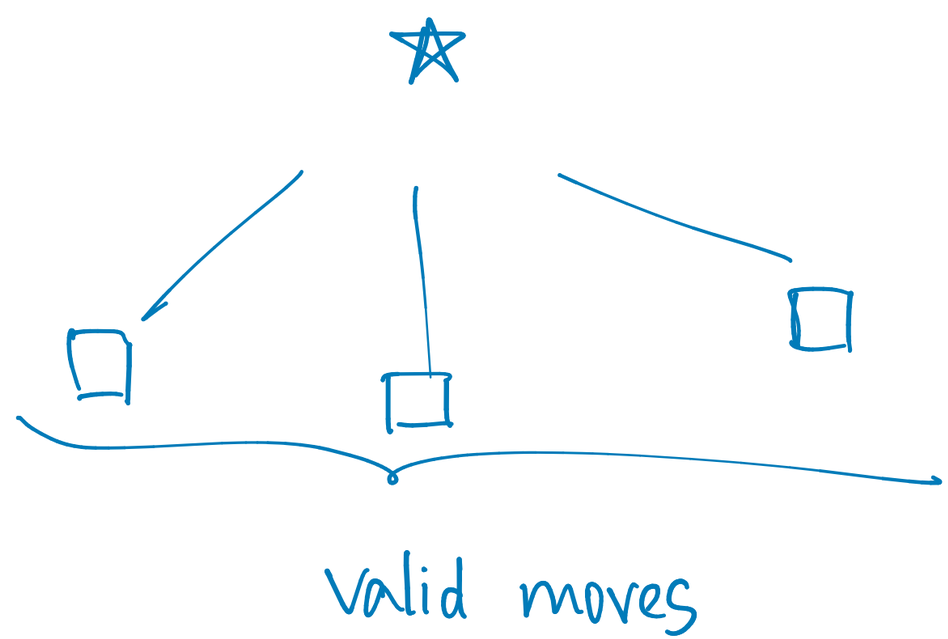
* assuming "optimal & intelligent" gameplay on both sides.

Game Tree


8
7
6
5
6
5
4
5
4
3
4
3
2
1
3
2
1
0
2
0
1
0
Other player win
Current player win

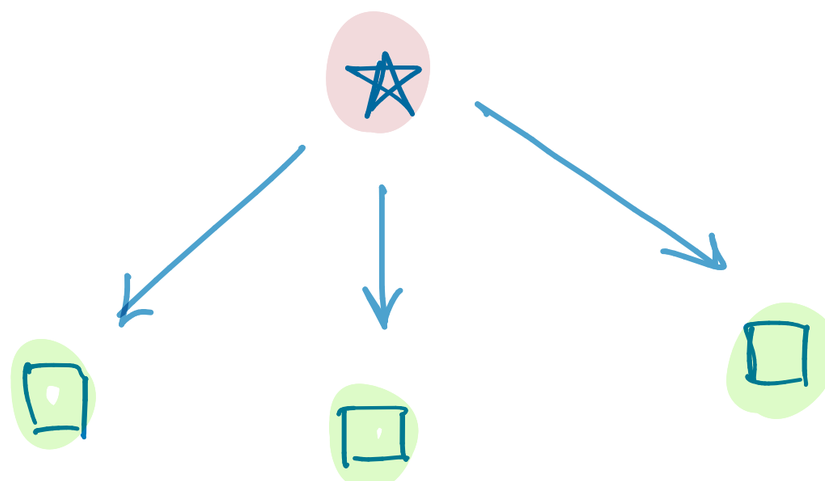
Nodes ⇝ positions of the game

Root ⇝ Starting point



Nodes ⇝ positions of the game

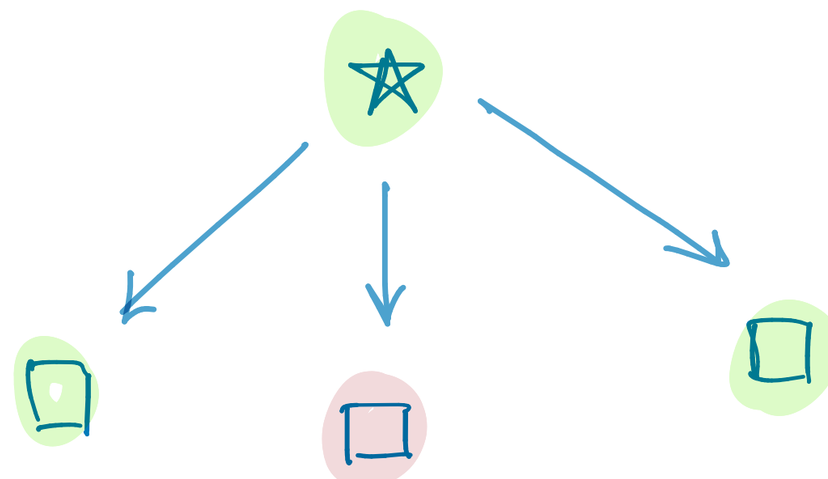
Root ⇝ Starting point



if all moves are wins for the current player
then the game is a win for the other player.

Nodes ⇝ positions of the game

Root ⇝ Starting point



if ≥ one move is a win for the other player

then the game is a win for the current player.

For identifying a node,

its enough to observe a child.

Before assigning a node,

we have to confirm that ALL children are .

Takeaways:

Game trees, winning positions.

recursive identification of wins

Food for thought:

Subtraction with other parameters.

Program that generates a game tree.