AI LAB ASSIGNMENT 04

Game Playing Agent | Minimax | Alpha-Beta Pruning

FAB FOUR

Ayush Arora | Aniket Yadav | Kailash Kumar | Avnit Kumar Anand
202051041 | 202051025 | 202051093 | 202051040

GAME TREE FOR NOUGHTS AND CROSSES

In the game of cross and noughts Initially all the places are empty. So, **Tree Size = 1**

Then there are total 9 possibilities for first move
Tree Size = (1 + 9)

Then for the next 9 states there are 8 possible moves . So, **Tree Size = (1 + 9 + 9x8)**

Then for the next 9*8 states, there are 7 possible moves. So, **Tree Size = (1 + 9 + 9x8 + 9x8x7)**

And it similarly it continues
.
.
.
.

So, **Tree Size = (1 + 9 + 9x8 + 9x8x7 + ... + 9x8x7x6x5x4x3x2x1)**

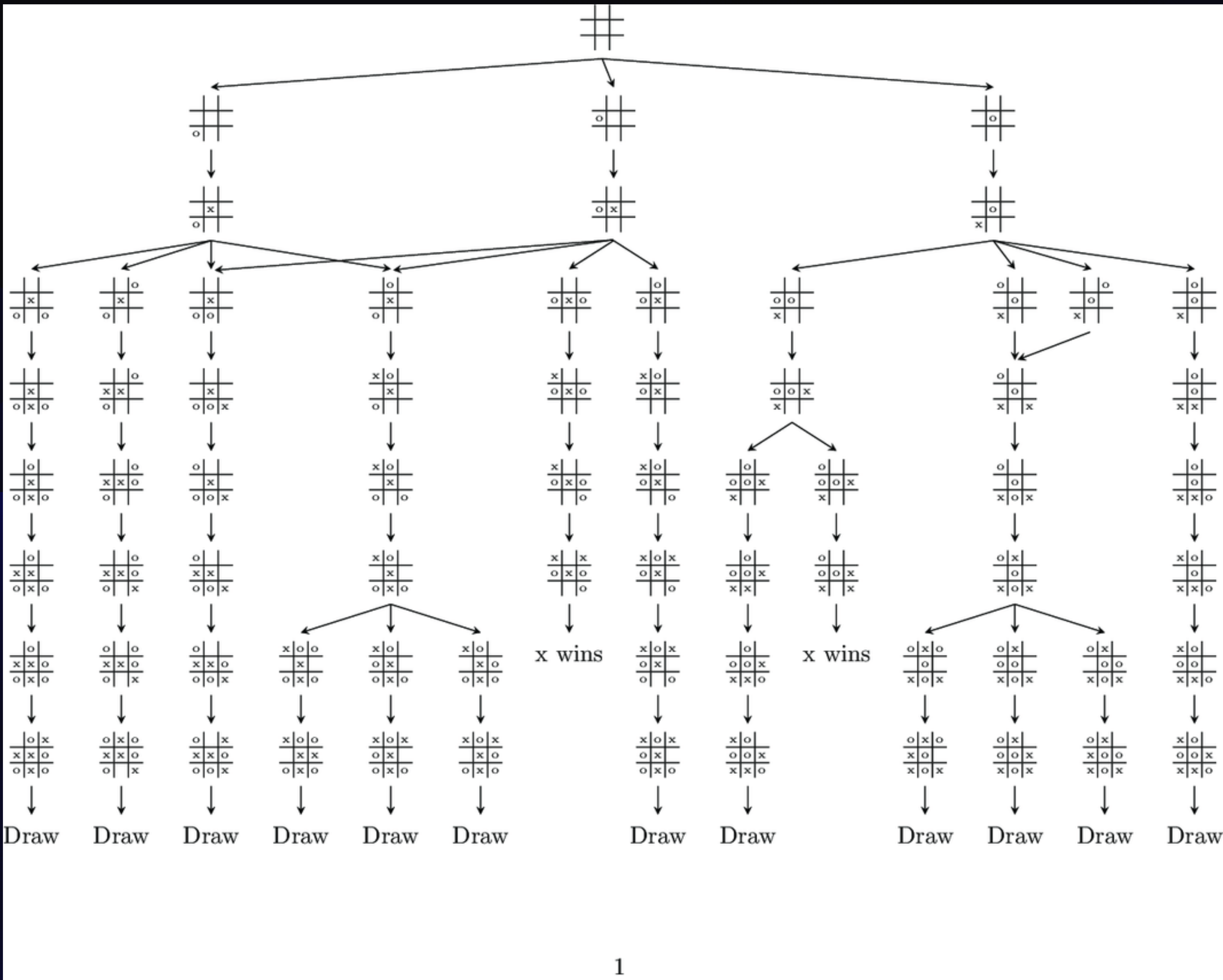
Tree Size = 9P0 + 9P1 + 9P2 +9P3 + ... +9P9
(npr = n!/r!)

Sum of the series is ≈ 9!*e   (e = 2.718...)

Hence ,**Tree Size = 9,86,410**

But actually the tree size is less as many states get terminated by winning before even 9 moves so we dont go futher in that matrix

So, ***Actual Size is around 549946.***



Nim game

In this game we have some objects arranged in different piles (3 piles in our case) and two players take turn to remove some objects from a pile.

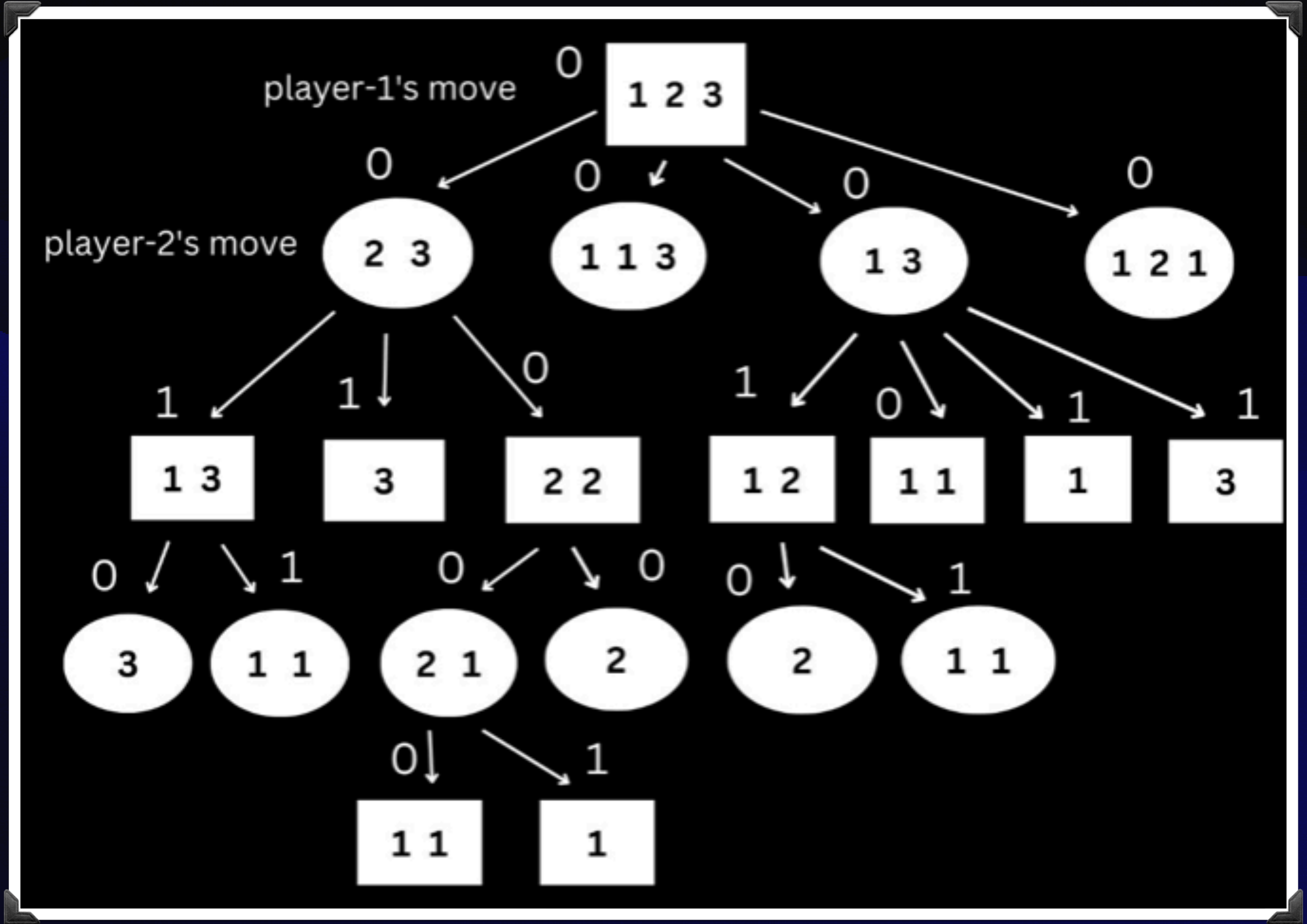
Rules

A player can remove any number of object but from a single pile.
The player left with no object to remove in their turn looses.

Some examples to show the minimax game tree

Ex. 1 Initial configuration is 1 2 3

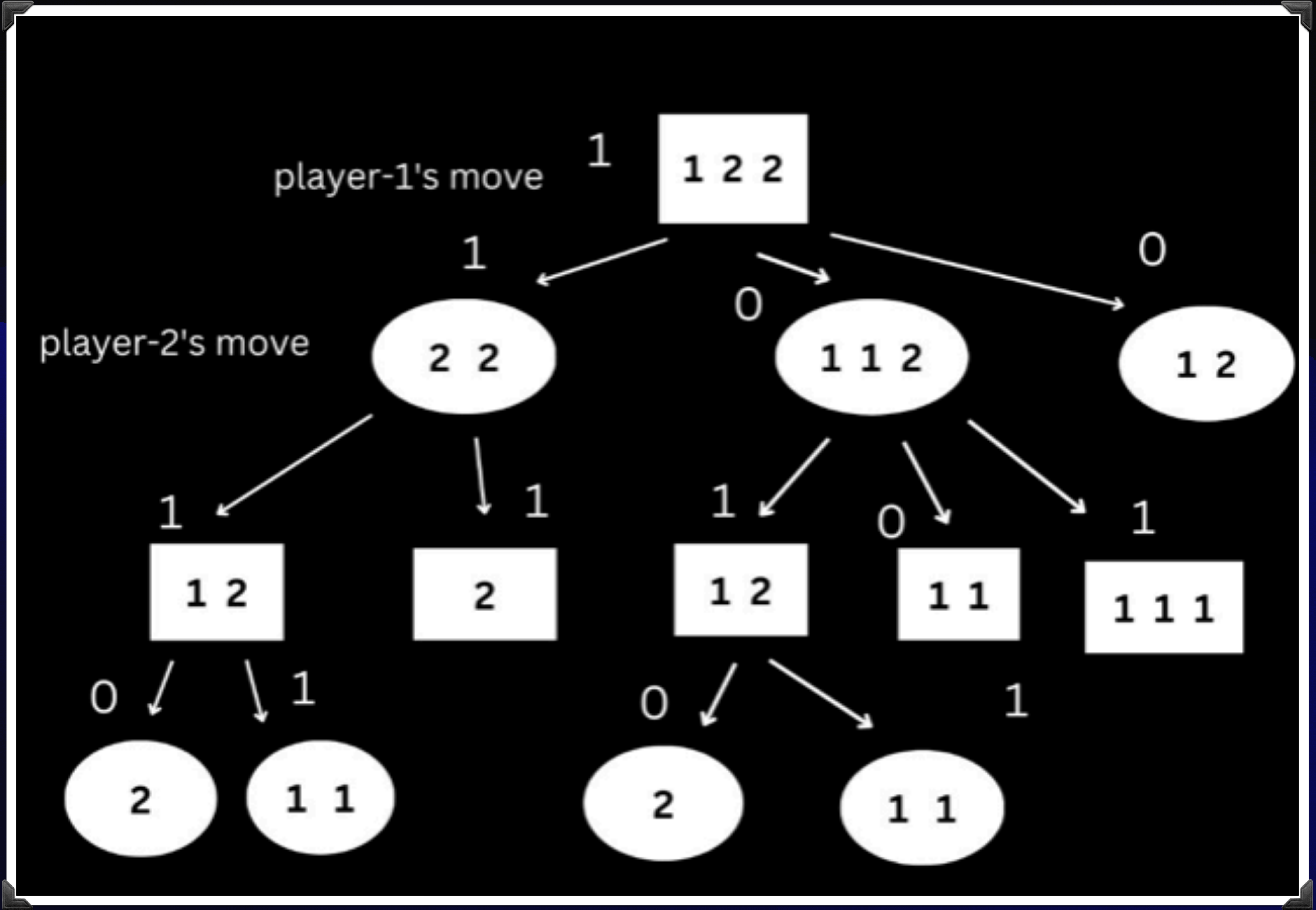
The player going first is the maximiser and needs a 1 to win.
And player two is the minimiser and needs 0 to win.



Player 2 wins.

Ex. 2 Initial configuration is 1 2 2

The player going first is the maximiser and needs a 1 to win.
And player two is the minimiser and needs 0 to win.



Player 1 wins.

Winning Strategy

The xor of number of objects in each pile is called the nim sum.
eg. if the piles are 1 2 3
nim sum = 1 xor 2 xor 3
hence nim sum = 0

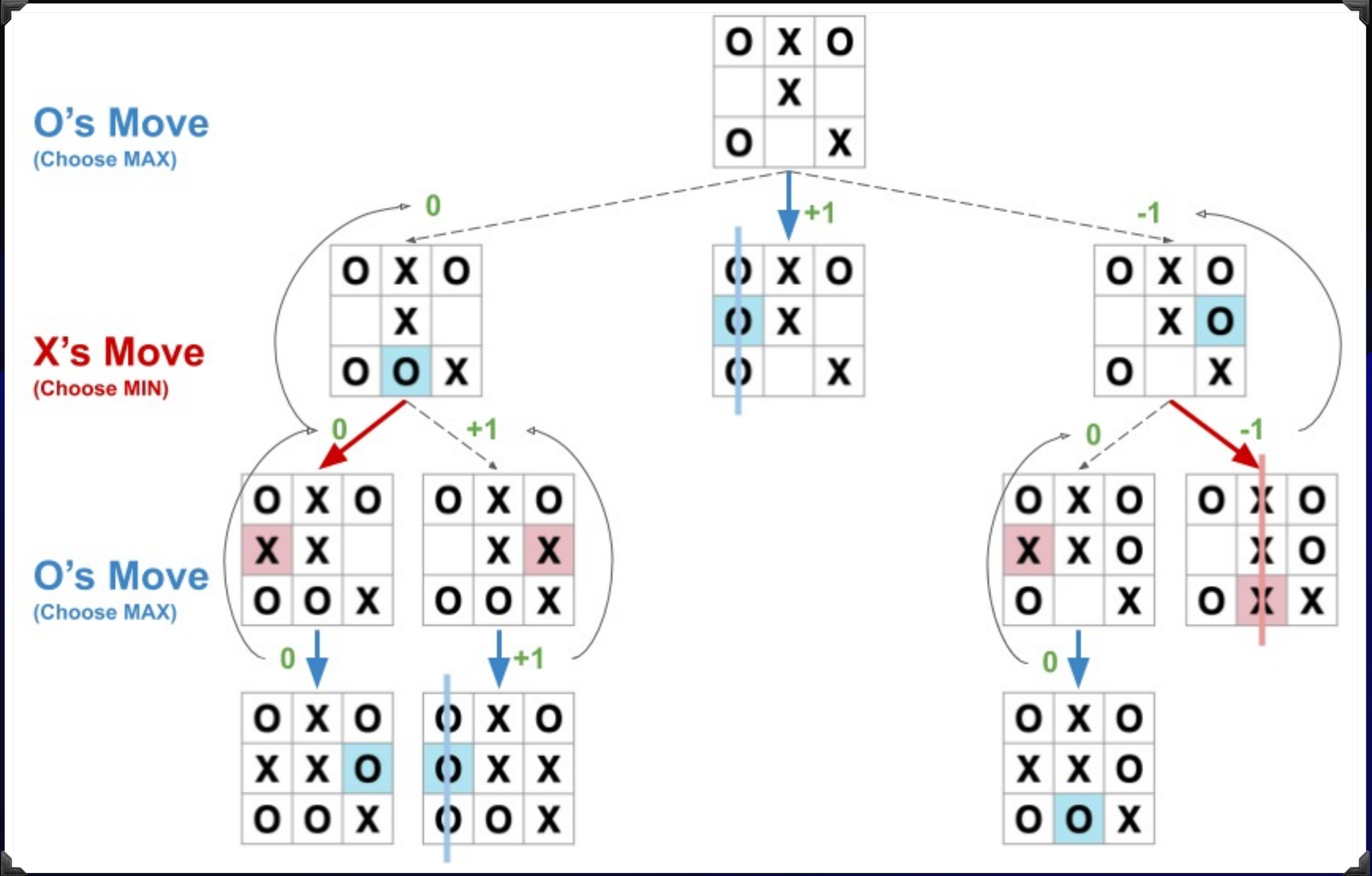
The strategy is that to win this game a player should make the nimsum zero in their turn.

eg. Piles: 1 2 4
nimsum = 7
we remove 1 from the third pile
Piles: 1 2 3
nimsum = 0

If the nim sum is already zero there is no possibility that we can make a move to make nim sum 0 in a single turn.
Hence if the nim sum is zero in the beginning player 2 will always win if played optimally else if nim sum is non zero in the begining then the player who goes first always wins.

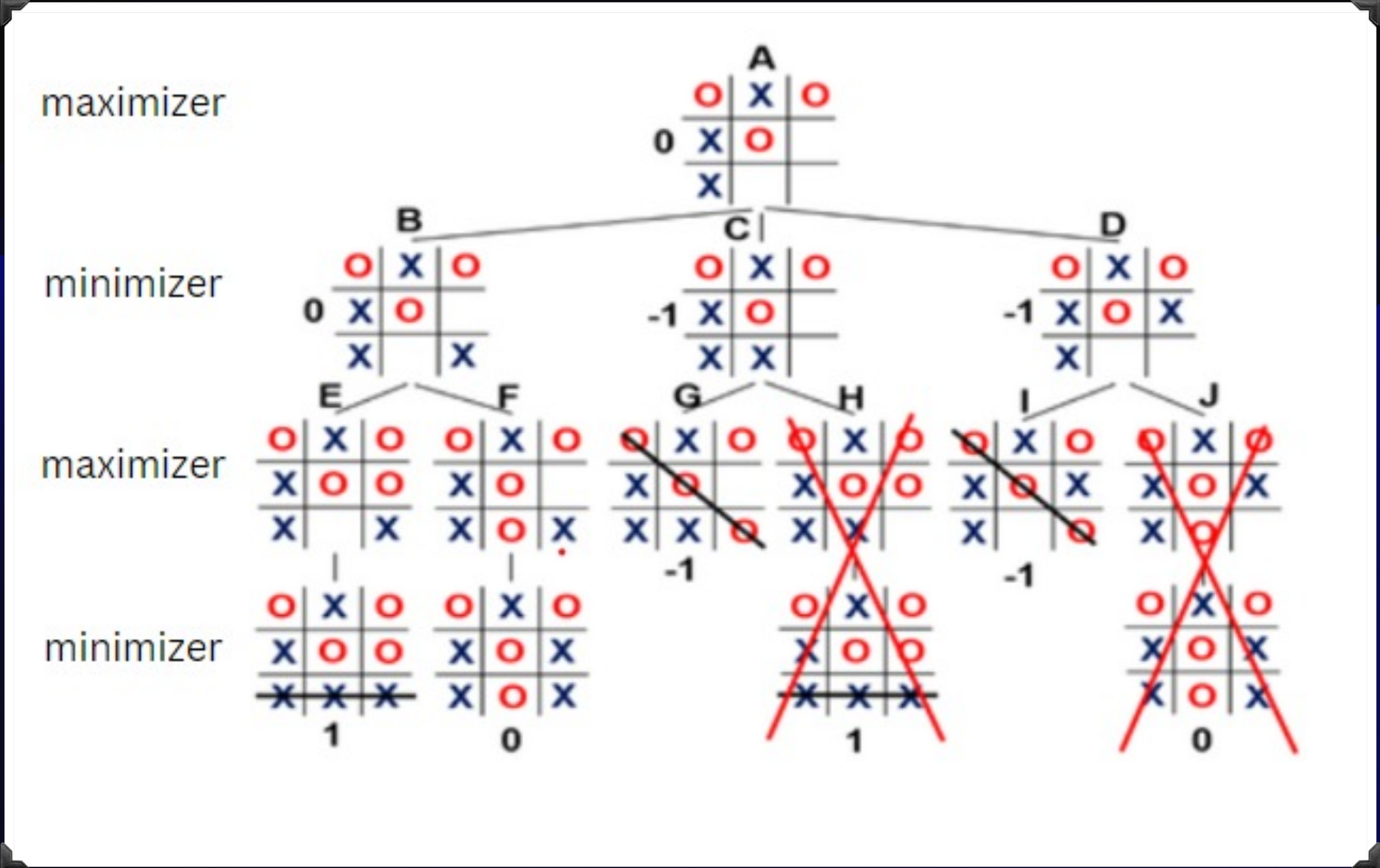
In the given Question no. 2 the nim sum is initially not zero
nim sum(10 , 7 , 9) = 4
Hence the player 1 can surely win if playes optimally.

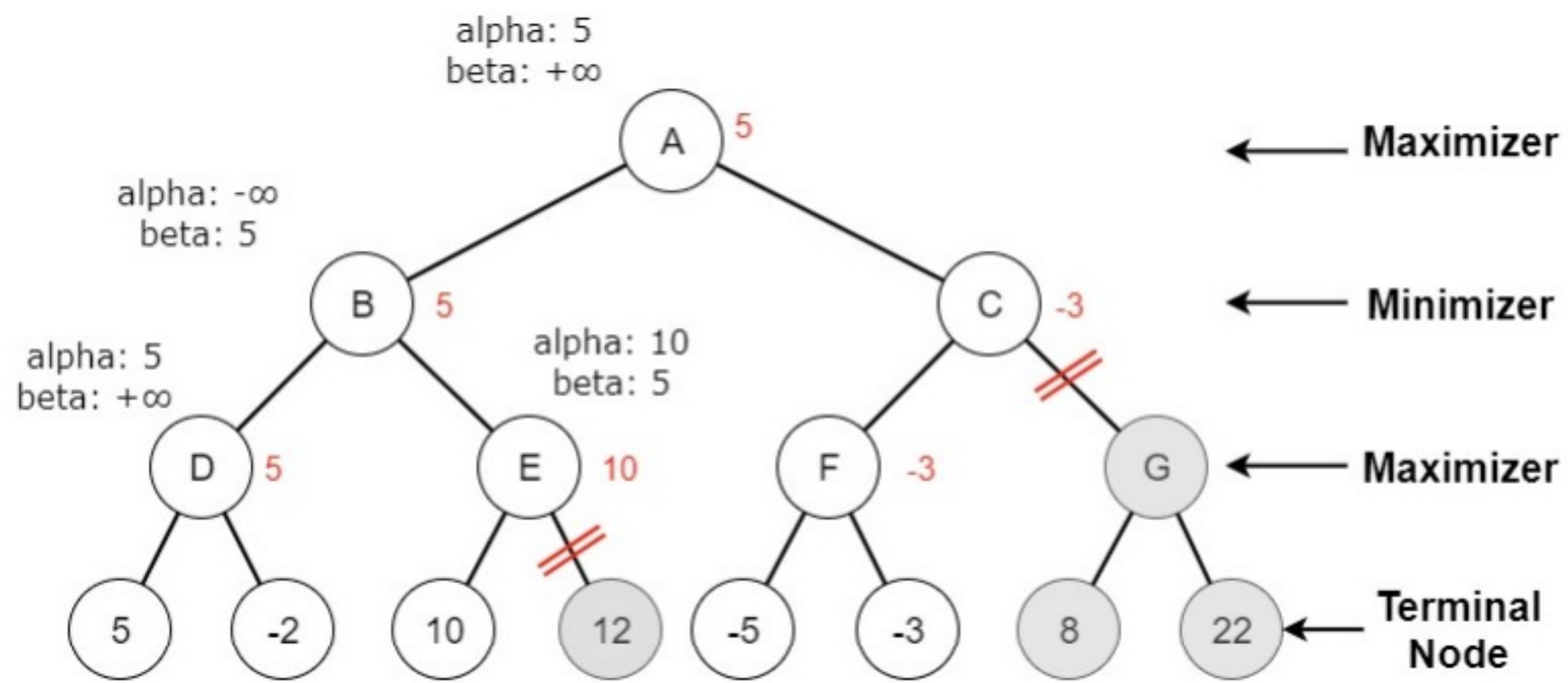
MINIMAX IN TIC-TAC-TOE



Alpha-Beta Pruning in Tic-Tac-Toe

Example




alpha: 5
beta: +∞
A
5
Maximizer
alpha: -∞
beta: 5
B
5
C
-3
Minimizer
alpha: 5
beta: +∞
alpha: 10
beta: 5
D
5
E
10
F
-3
G
Maximizer
5
-2
10
12
-5
-3
8
22
Terminal
Node

Alpha-Beta Pruning Time Complexity

The alpha-beta pruning time complexity can be shown using a recurrence relation as follows:

$$T(m) = b * T(m-1) + O(1)$$

where $T(m)$ is the time complexity of the algorithm at depth m, b is the effective branching factor, and $O(1)$ is the constant time to evaluate each node.

Assuming perfect ordering of leaf nodes, at depth m, the algorithm will only need to evaluate b^(m/2) nodes, as the alpha-beta pruning mechanism will eliminate half of the nodes at each level.

Thus, the time complexity at depth m can be expressed as:

$$T(m) = b^{(m/2)} * O(1) + T(m-1)$$

Using this recurrence relation, we can calculate the overall time complexity as follows:

$$\begin{gathered} T(m) = b^{(m/2)} * O(1) + T(m-1) \\ = b^{(m/2)} * O(1) + b^{((m-1)/2)} * O(1) + T(m-2) \\ = \dots \\ = b^{(m/2)} * O(1) + b^{((m-1)/2)} * O(1) + \dots + b^{(2/2)} * O(1) + b^{(1/2)} * O(1) + T(0) \\ = O(b^{(m/2)}) \end{gathered}$$

Thus, under perfect ordering of leaf nodes, the alpha-beta pruning time complexity is ***O(b^(m/2)***).

References

[1]A first course in Artificial Intelligence, Deepak Khemani (Chapter 8)

[2] Artificial Intelligence: a Modern Approach, Russell and Norvig (Fourth edition)

[3] https://youtu.be/SzziJsmVhzY

[4] https://youtu.be/Vus2I9NVNzc

THANK YOU

:)