CMSC 170: Games

Min-Max Algorithm

Katherine Loren M. Tan
Institute of Computer Science
University of the Philippines Los Baños

LEARNING OUTCOMES

At the end of the session, the students should be able to:

- understand the Min-Max Algorithm;
- understand the use of Alpha and Beta Pruning ; and
- implement min-max algorithm and alpha beta pruning in a 2-player deterministic game.

Games

Every game has different sets of properties.

Games

We will discuss 2-player deterministic games and how to make an AI agent.

States

We can define it as S as the set of states.

Ex. S= $\{S_0\}$ //start of the game

Player

There are also P as the of players.

Ex. $P=\{P_1,P_2\}$

Result

There are also results in games. Result(s,a) -> s'

Results(s,a) -> s’

s’ are the new states made when action a is done to a state s.

Actions(s,p)

provides the set of possible actions/moves for a player.

Terminal

It returns true when the state is terminal or there is an end result or false otherwise.

Utility(s,p)

Returns the current value of the state s to a player p.

Utility

It is expresses the value of the current game state to the player. It is usually expressed as +/- numbers or 0’s or 1’s.

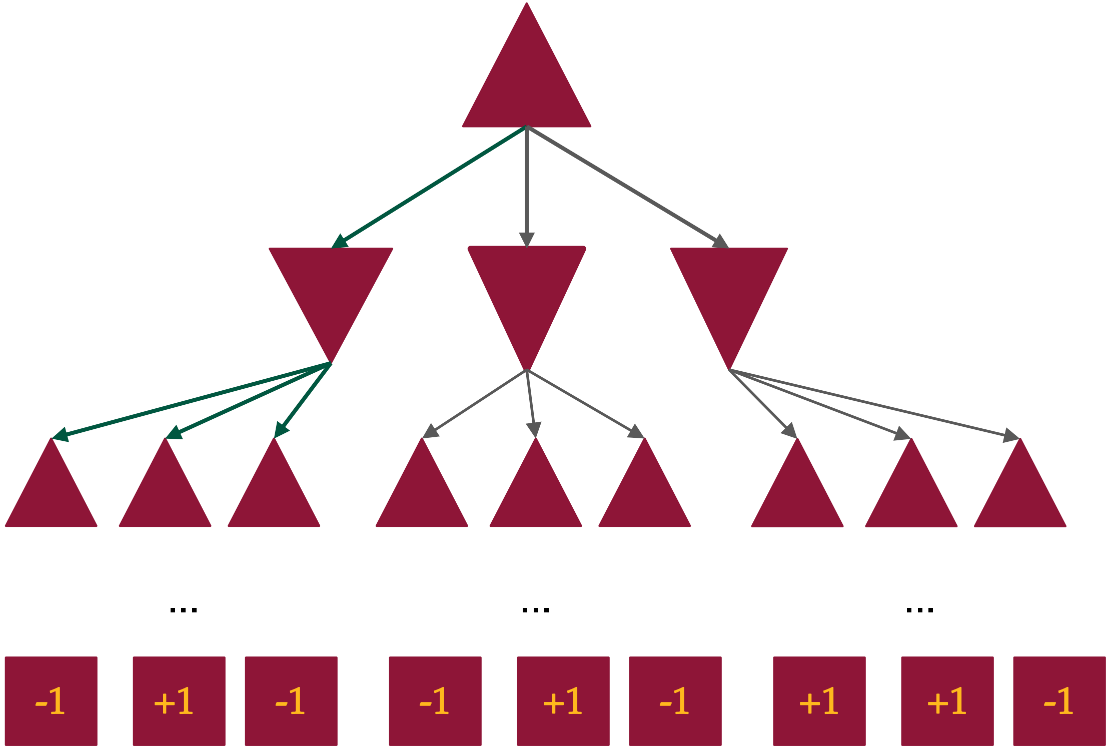
Zero-Sum Games

These are games where the overall sum of the utilities for both the players is 0.

Zero-Sum Games

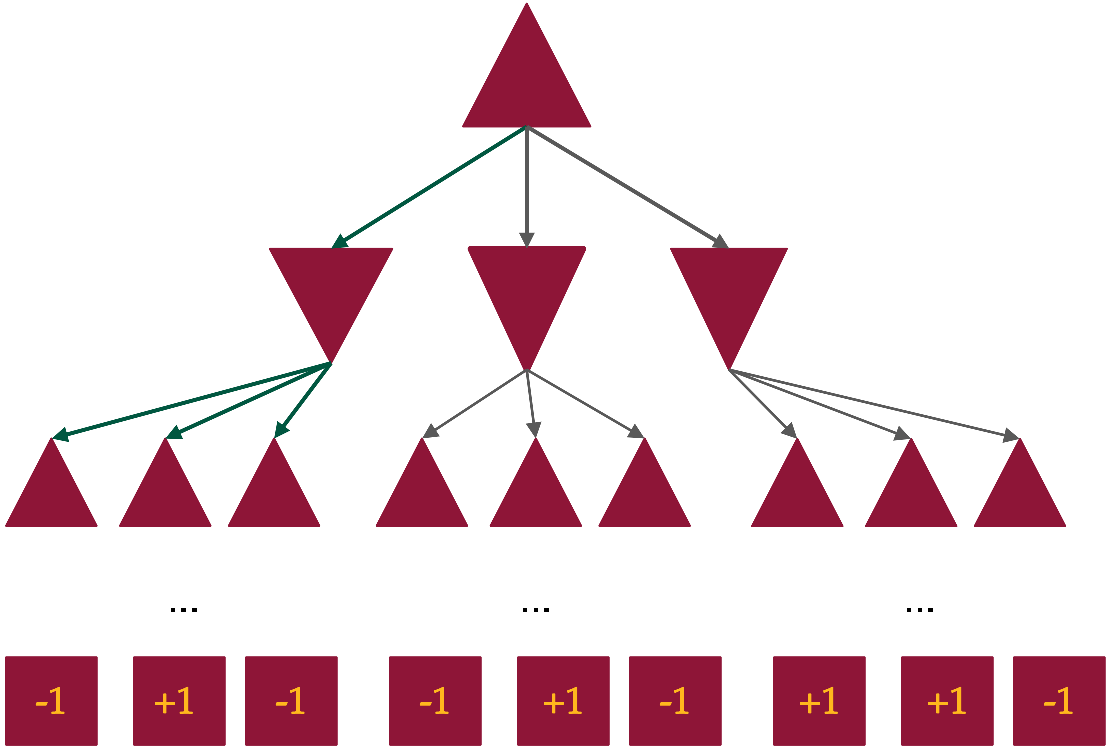
Ex. The utility for a win is 1 and for the loss is -1.

MINMAX ALGORITHM



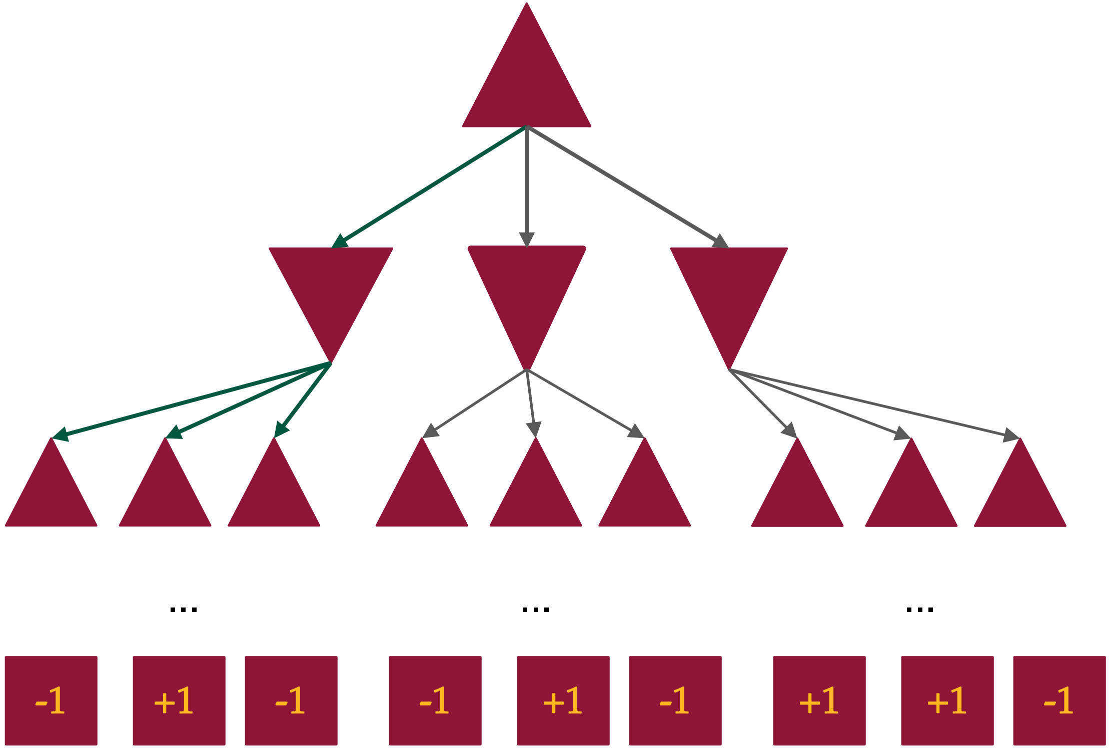
MINMAX ALGORITHM

Game Tree



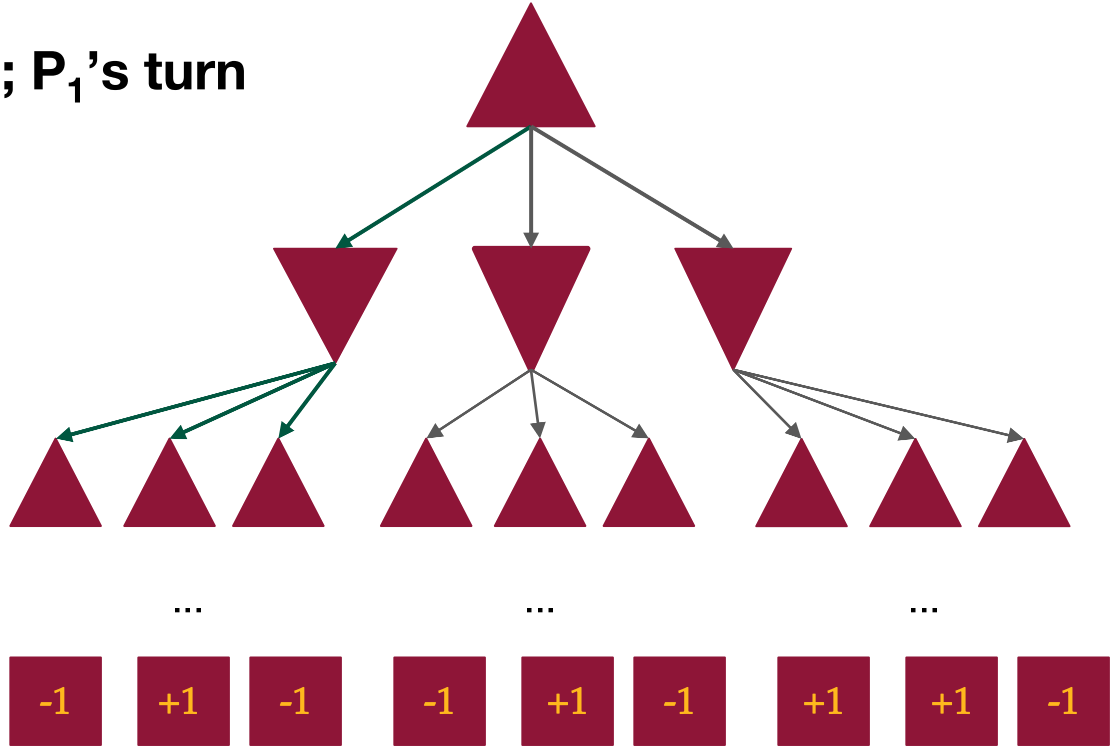
MINMAX ALGORITHM

Game Tree

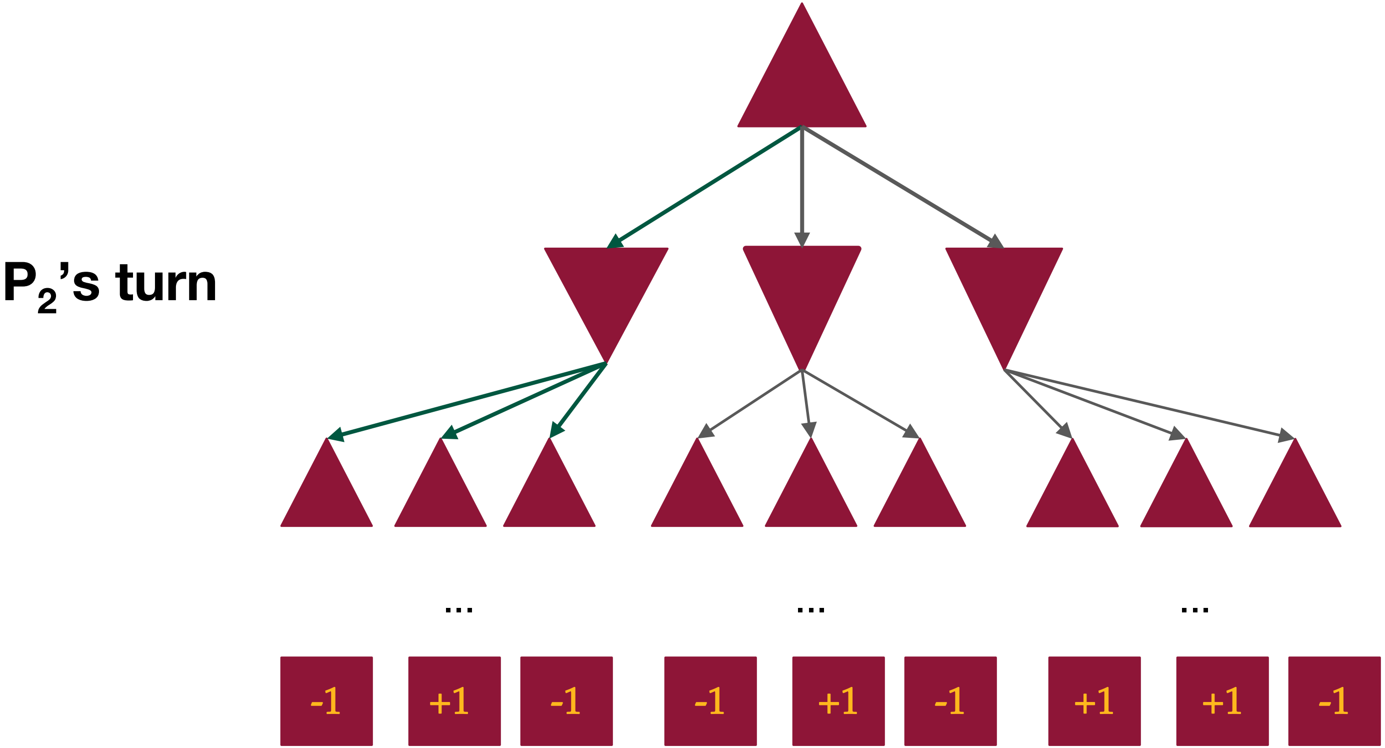


MINMAX ALGORITHM

Start of game; P_1's turn

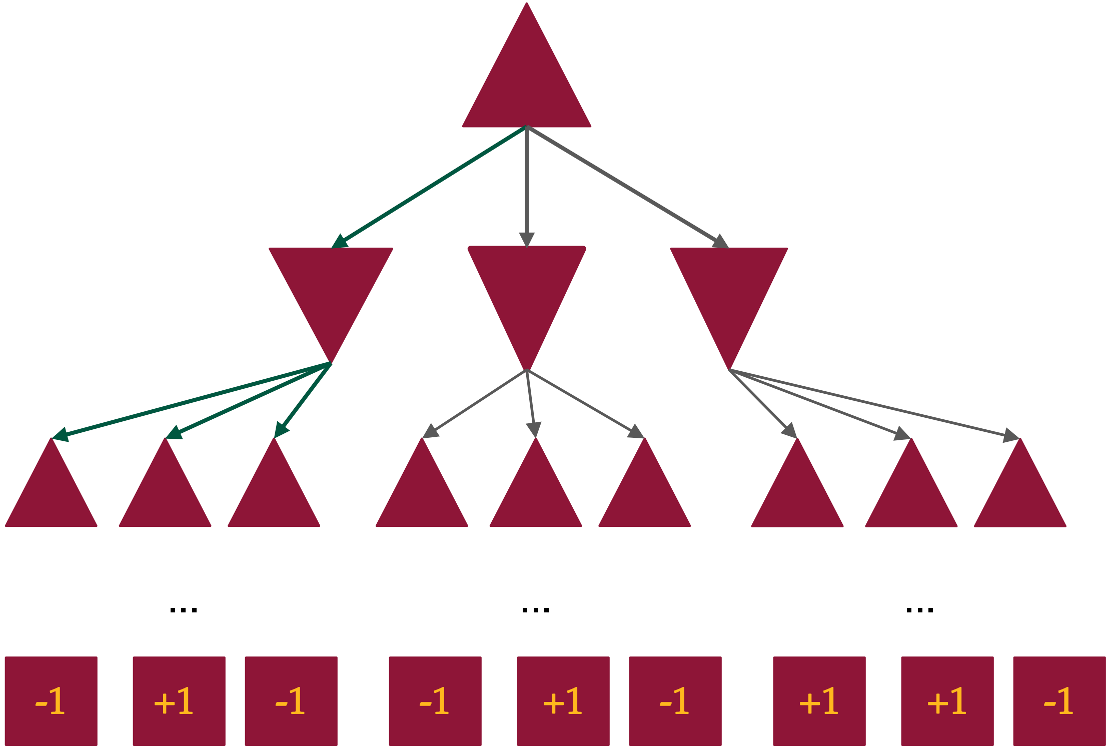


MINMAX ALGORITHM

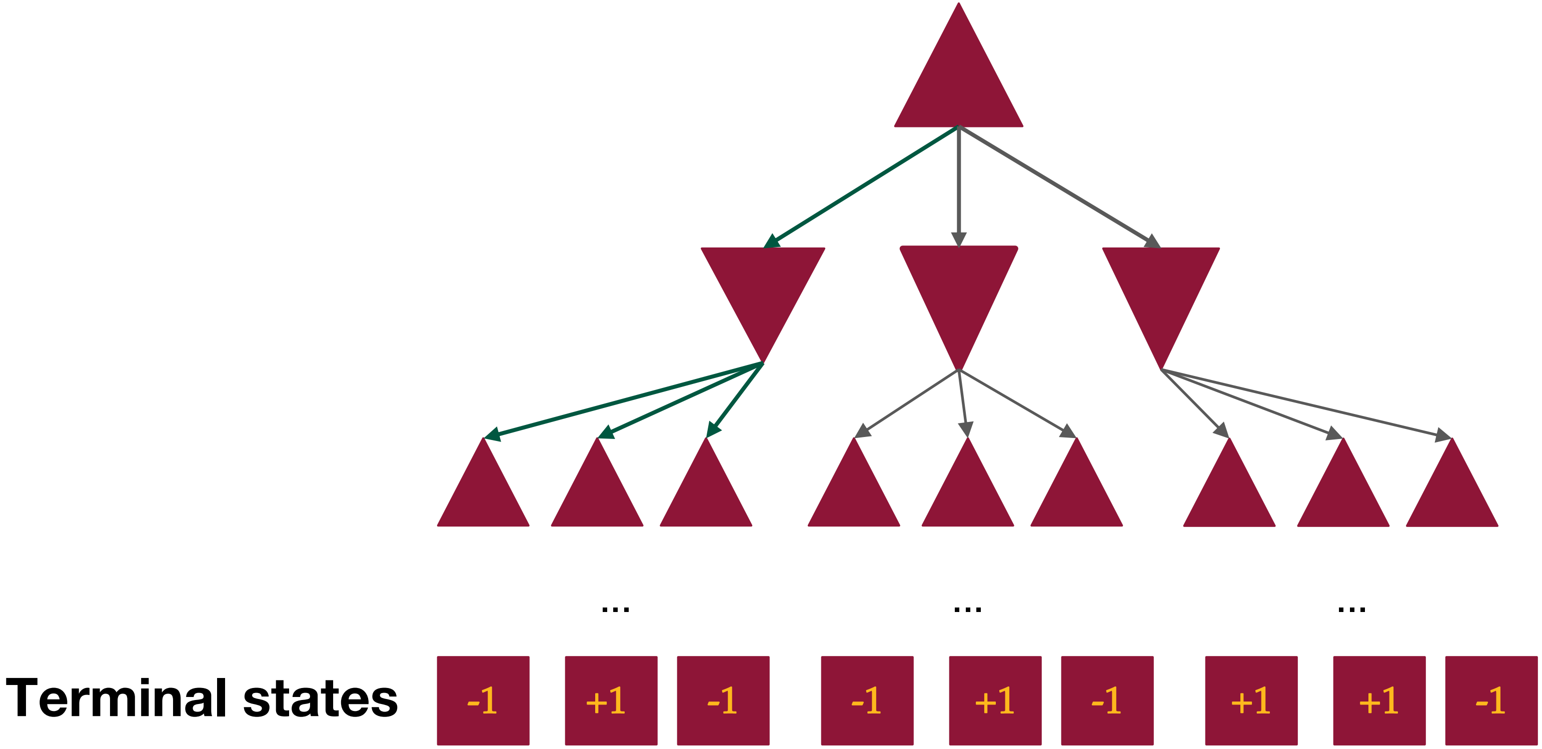


MINMAX ALGORITHM

P_1's turn again



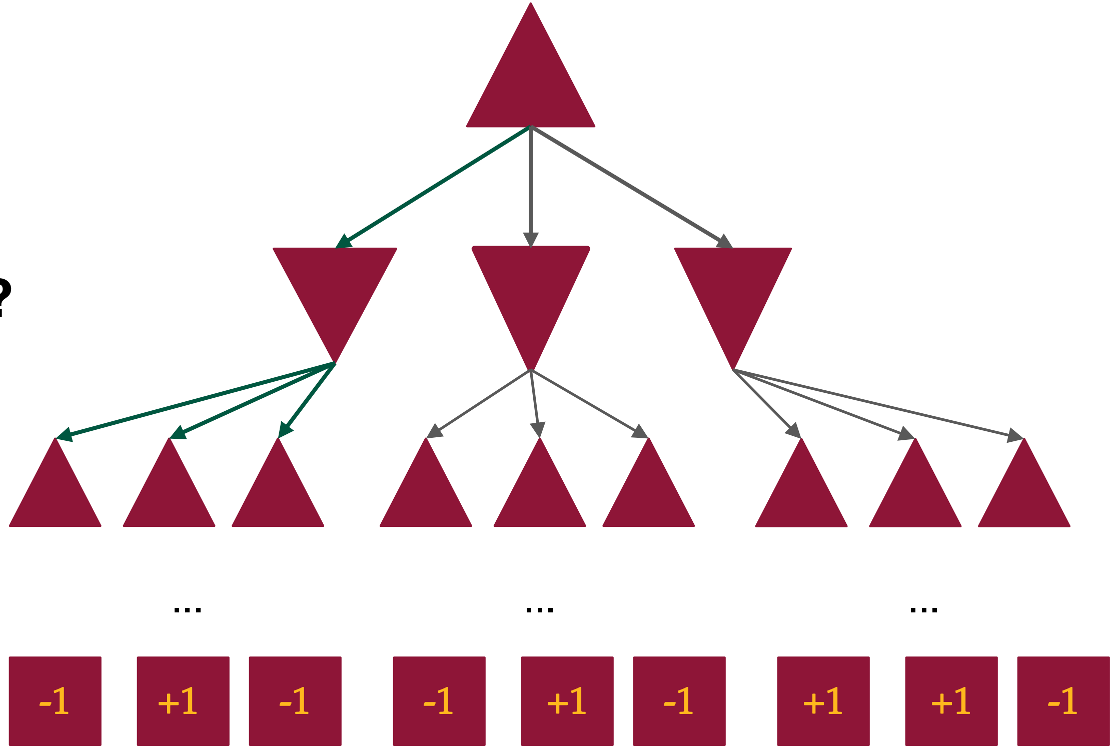
MINMAX ALGORITHM



Each node has a utility value. **P_1** attempts to **maximize** the values; **P_2** attempts to **minimize** the value thus resulting to an adversarial environment.

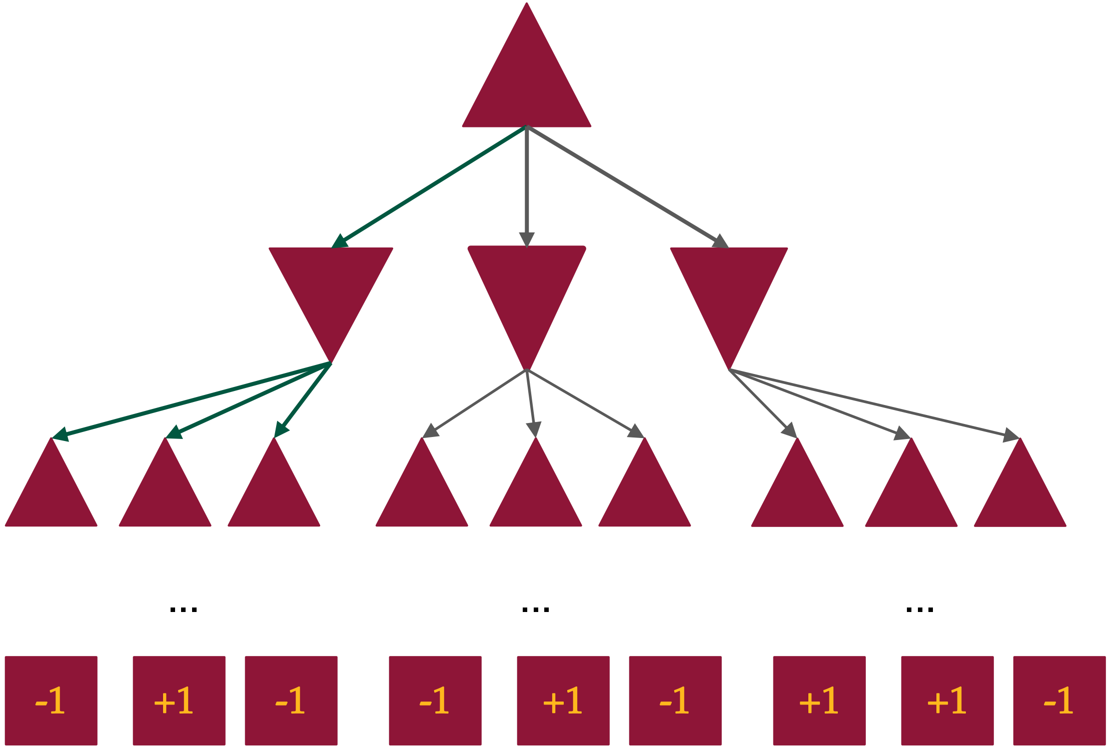
MINMAX ALGORITHM

How to get utility of the inner nodes?



MINMAX ALGORITHM

Use the terminal states to Return the values by following the min-max behavior of P_1 and P_2



value (s):

if s □ is : utility(s)
if s △ is : maxValue(s)
If s ▽ is : minValue(s)

```
maxValue(s):
        m = -∞
for a, s' in successors(s):
        v = value(s')
        m = max(m,v)
    return m
```

```
minValue(s):
        m = +∞
    for a, s′ in successors(s):
        v = value(s′)
        m = min(m,v)
    return m
```

It has a time complexity of

$O(B^m)$

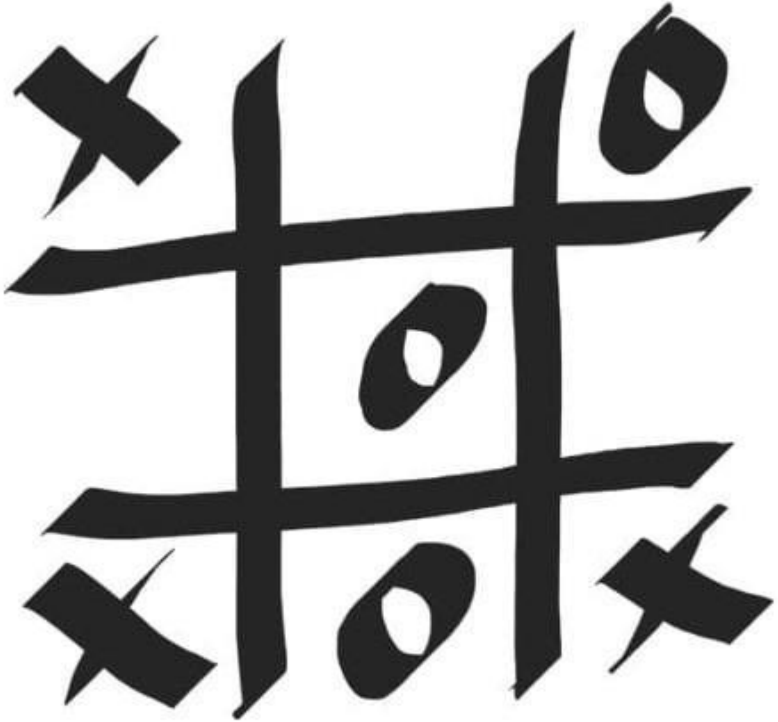
BRANCHING FACTOR

approximate number of choice (successors) a player has at each node.

DEPTH

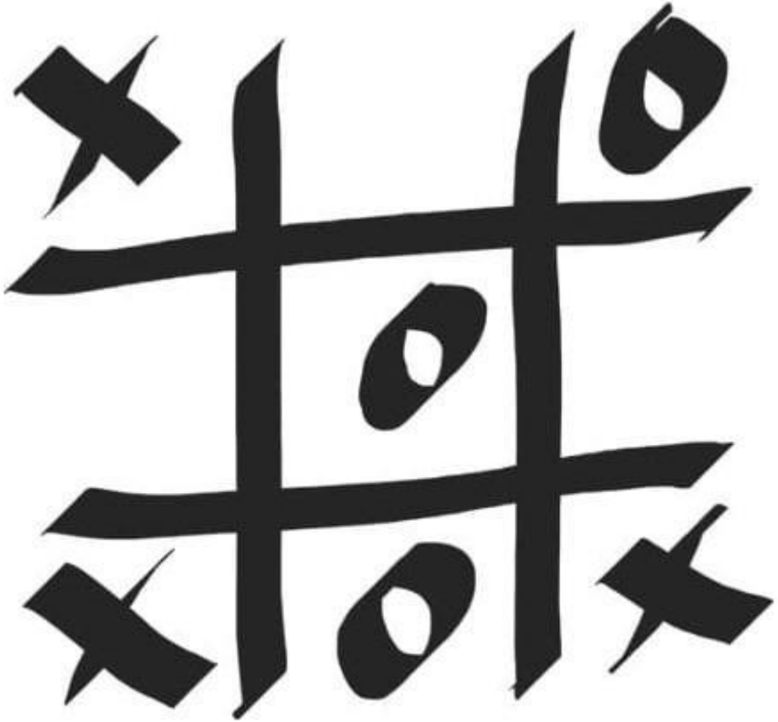
total number of turns for P_1 and P_2 to reach a terminal state

This is our current state s.



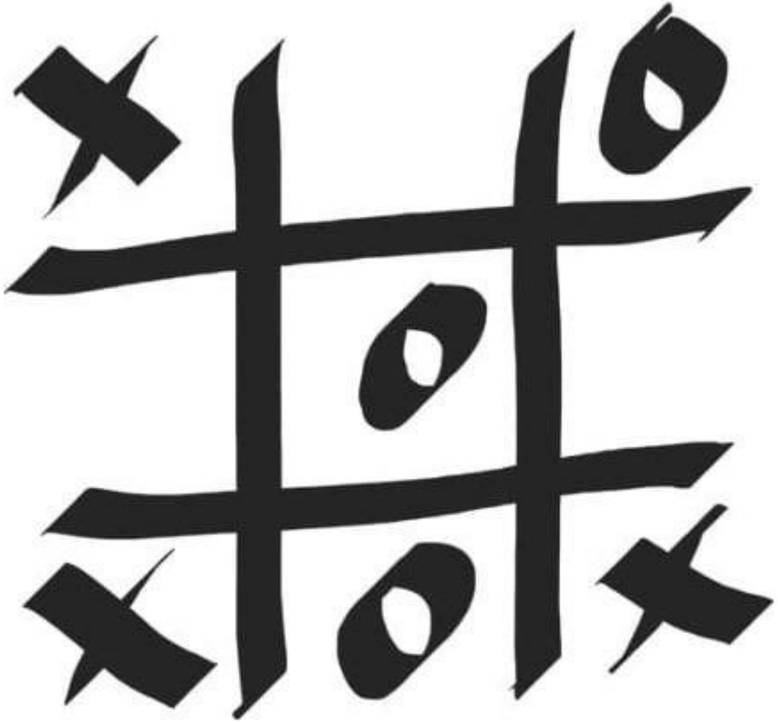
TICTACTOE GAME

We set X as the maximizer and O as the minimizer.



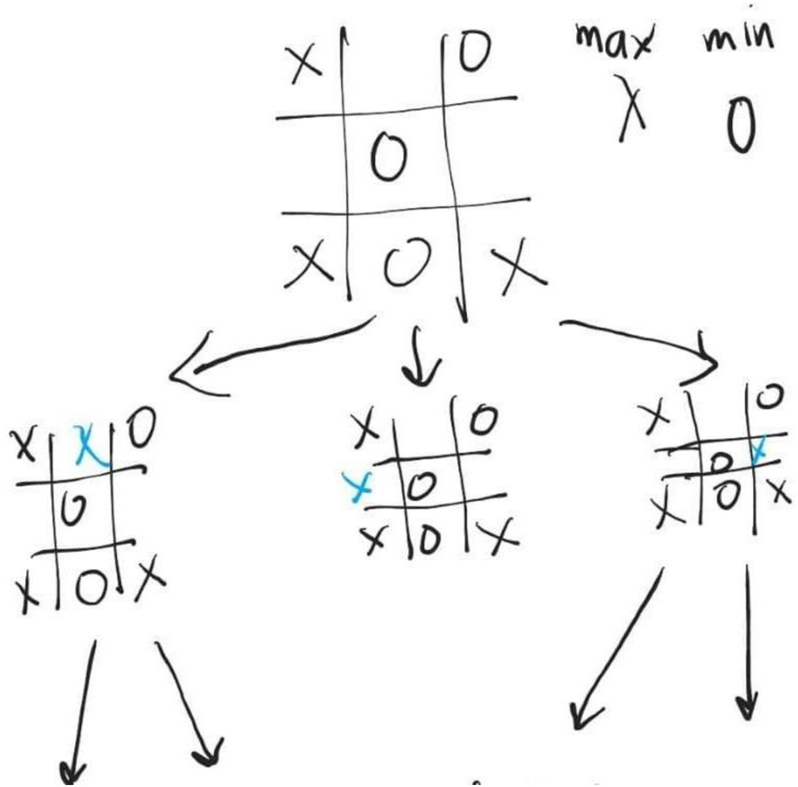
TICTACTOE GAME

Its X's Turn



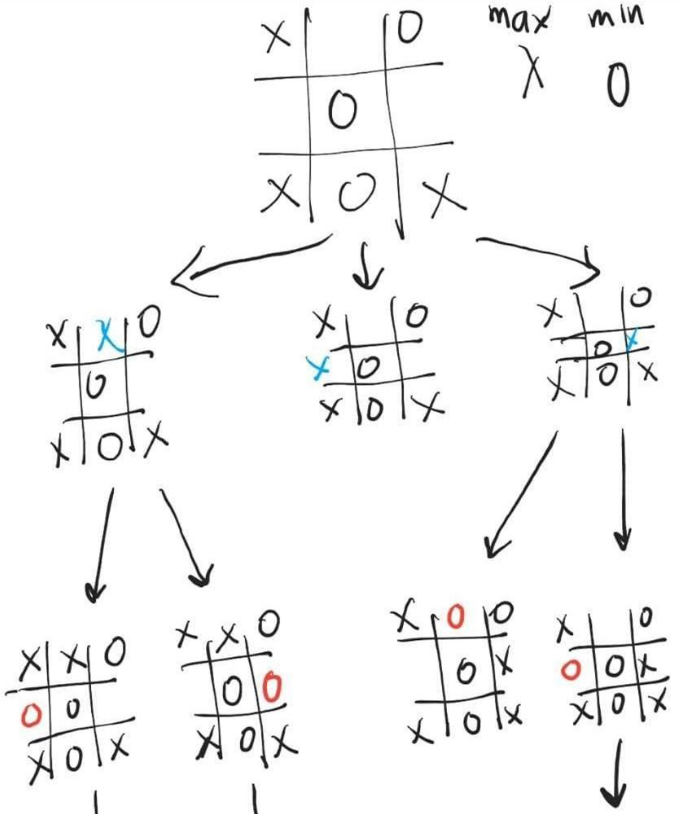
TICTACTOE GAME

Its X's Turn



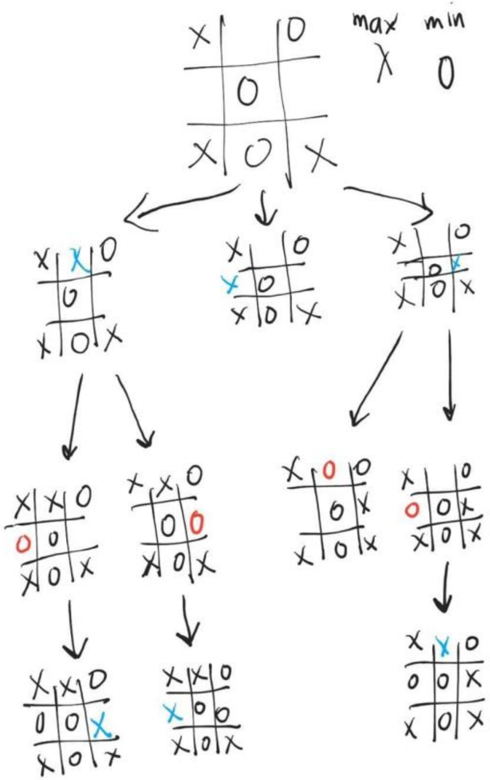
TICTACTOE GAME

Its 0's Turn



TICTACTOE GAME

Its X’s Turn



ALPHA-BETA PRUNING

The algorithm used to avoid searching subtrees of moves that won’t be considered.

ALPHA(α)

The best solution (lower bound) for the maximizer on the path to the root.

BETA(β)

The best solution (upper bound) for the minimizer on the path to the root.

value (s, α ,β):

if s $\square$ is : utility(s)

if s $\triangle$ is : maxValue(s,α,β)

If s $\triangledown$ is : minValue(s, α,β)

```
maxValue(s,α,β):
        m = -∞
    for a, s′ in successors(s):
        v = value(s′,α,β)
        m = max(m,v)
        if v>= β:
            return m
        α = max(α,m)
    return m
```

```
minValue(s,α,β):
        m = +∞
    for a, s′ in successors(s):
        v = value(s′, α,β)
        m = min(m,v)
        if v<=  : α
            return m
        β = min(β,m)
    return m
```

For questions and inquiries, you can email me at

kmtan4@up.edu.ph

EXERCISE on MinMax Algorithm

Create a tictactoe program with a MinMax AI agent. It must have a user interface. It must ask whether the user or the AI would go first.

SOME REMINDERS:

- Naming convention for exercise: surname_minmax.
- Python or Java can only be used for the exercise.
- Do not forget to put a journal in your ReadMe file in Github.
- Lastly, **Honor and Excellence.**