

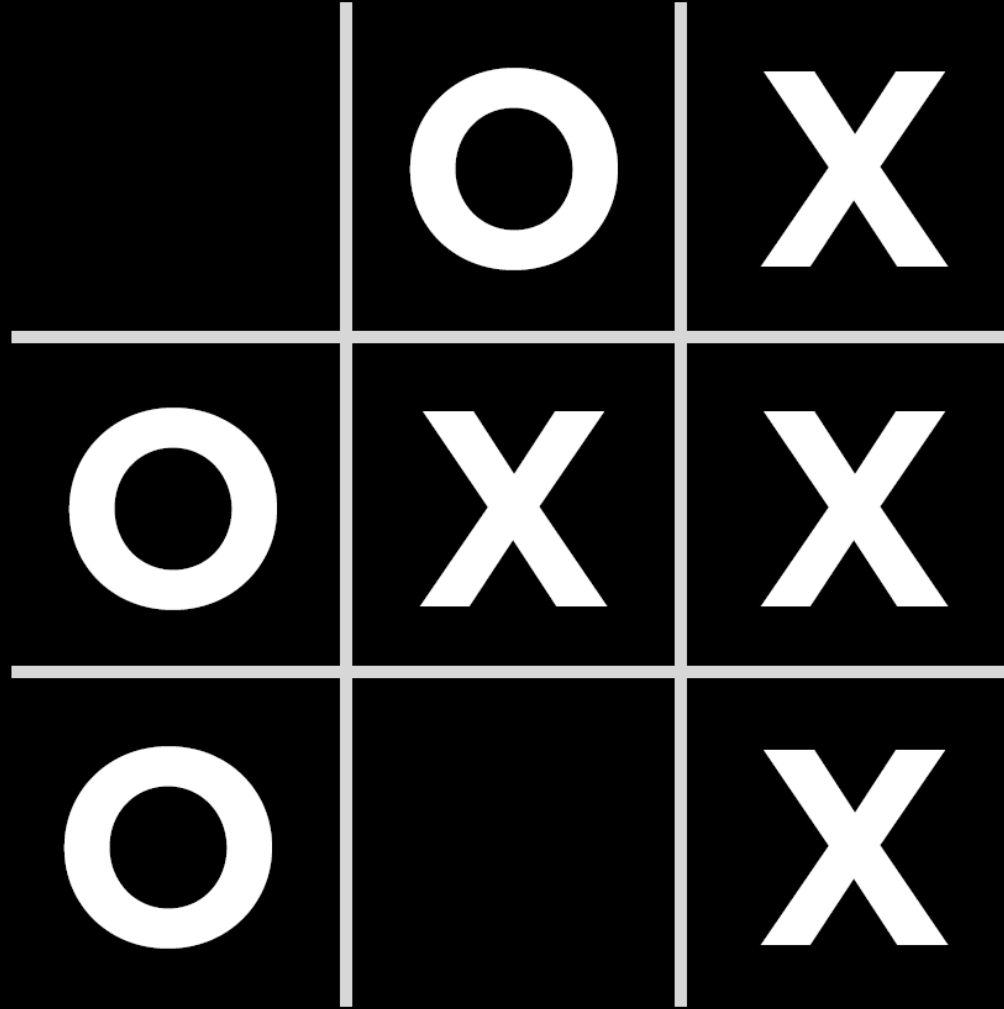
FUNDAMENTOS DE INTELIGÊNCIA ARTIFICIAL

Aula 6: MinMax

Prof. Dr. Rodrigo Xavier de Almeida Leão
Cientista de Dados e Big Data



BUSCA COM ADVERSÁRIO



BUSCA COM ADVERSÁRIO

- SEMELHANTE A UMA BUSCA

EXISTEM:

- AÇÕES
- CENÁRIOS
- TRANSIÇÃO

	O	X
O	X	X
O		X

BUSCA COM ADVERSÁRIO

PROCURA-SE O MELHOR CAMINHO, MAS EXISTE UM ADVERSÁRIO TENTANDO EVITAR ESTE RESULTADO.

	O	X
O	X	X
O		X

ALGORITMO MINMAX

- **COMO EXPRESSAR O DESEJO DE GANHAR PARA O COMPUTADOR?**
- **COMO TRADUZIR EM NÚMEROS?**
- **DEVEMOS DAR VALOR A CADA ESTADO POSSÍVEL DO JOGO.**

ALGORITMO MINMAX

O	X	X
O	O	
O	X	X

-1

X	O	X
O	O	X
X	X	O

0

O		X
	X	O
X	O	X

1

BUSCA COM ADVERSÁRIO

JOGADOR X -> MAX PLAYER

TENTA MAXIMIZAR O VALOR DO JOGO

JOGADOR O -> MIN PLAYER

TENTA MINIMIZAR O VALOR DO JOGO

O	X	X
O	O	
O	X	X

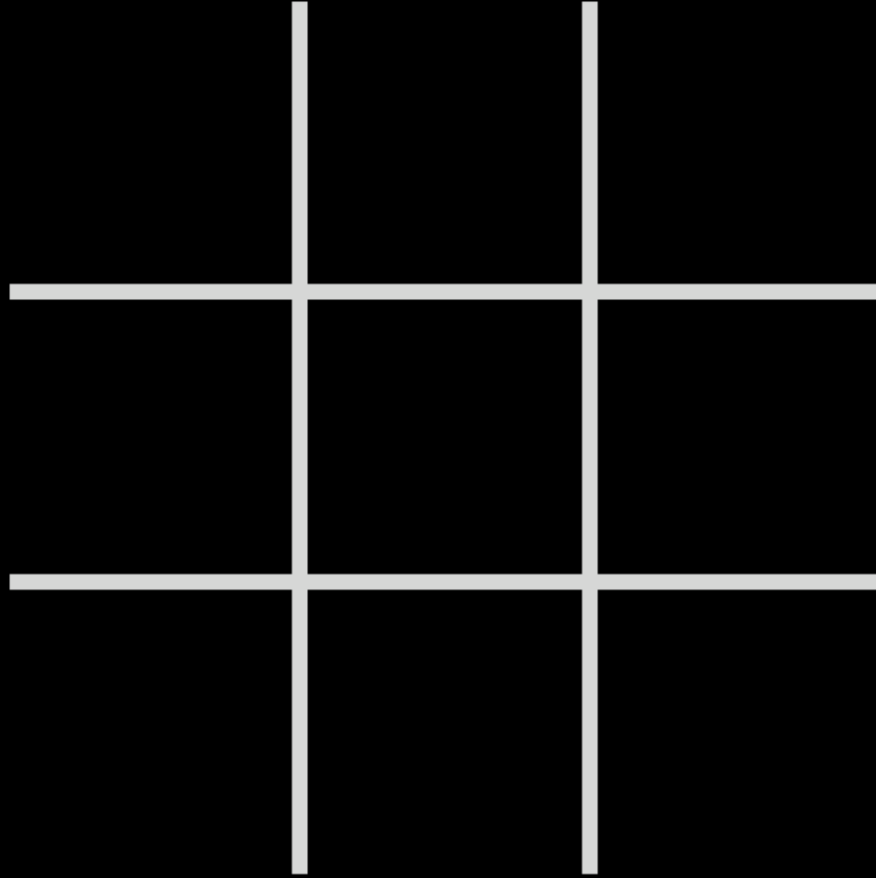
X	O	X
O	O	X
X	X	O

O		X
	X	O
X	O	X

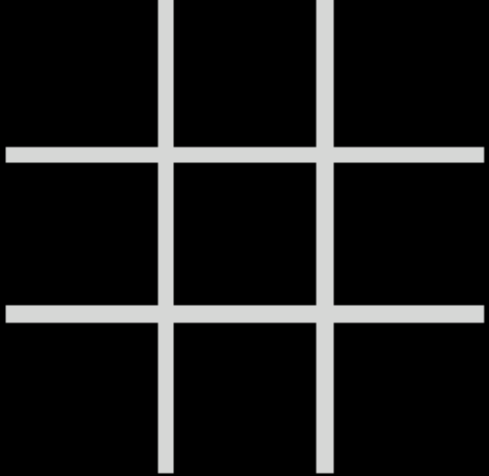
PLANEJANDO O ALGORITMO JOGO

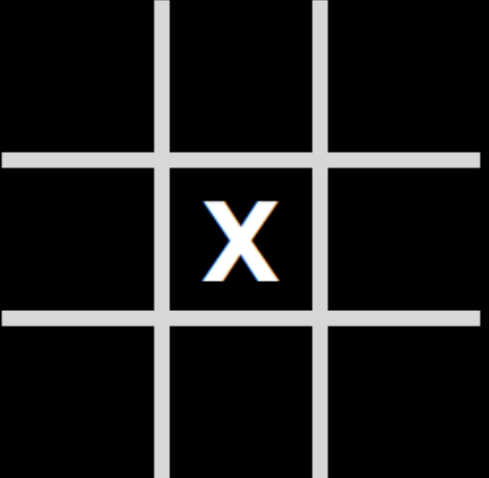
1. **$S \rightarrow$ Estado do jogo $\rightarrow S_0 \rightarrow$ Estado inicial**
2. **$\text{Player}(S) \rightarrow$ Retorna quem joga no estado S**
3. **$\text{Ação}(S) \rightarrow$ Retorna as ações possíveis no estado S**
4. **$\text{Resultado}(S, A) \rightarrow$ Retorna o resultado da Ação ' a ' no estado ' S '**
5. **$\text{Terminal}(S) \rightarrow$ Indica se o estado S é terminal (fim do jogo)**
6. **$\text{Utilidade}(S) \rightarrow$ Atribui um valor numérico ao estado terminal**

Initial State



PLAYER(*s*)

PLAYER() = **X**

PLAYER() = **O**

ACTIONS(s)

$$\text{ACTIONS}\left(\begin{array}{|c|c|c|} \hline & \text{X} & \text{O} \\ \hline \text{O} & \text{X} & \text{X} \\ \hline \text{X} & & \text{O} \\ \hline \end{array} \right) = \left\{ \begin{array}{|c|c|c|} \hline \text{O} & & \\ \hline & & \\ \hline & & \\ \hline \end{array} , \begin{array}{|c|c|c|} \hline & & \\ \hline & & \\ \hline & \text{O} & \\ \hline \end{array} \right\}$$

RESULT(s, a)

$$\text{RESULT}\left(\begin{array}{|c|c|c|} \hline & \text{X} & \text{O} \\ \hline \text{O} & \text{X} & \text{X} \\ \hline \text{X} & & \text{O} \\ \hline \end{array}, \begin{array}{|c|c|c|} \hline \text{O} & & \\ \hline & & \\ \hline & & \\ \hline \end{array}, \right) = \begin{array}{|c|c|c|} \hline \text{O} & \text{X} & \text{O} \\ \hline \text{O} & \text{X} & \text{X} \\ \hline \text{X} & & \text{O} \\ \hline \end{array}$$

TERMINAL(s)

$$\text{TERMINAL}\left(\begin{array}{c|c|c} \text{o} & & \\ \hline \text{o} & \text{x} & \\ \hline \text{x} & \text{o} & \text{x} \end{array}\right) = \text{false}$$

$$\text{TERMINAL}\left(\begin{array}{c|c|c} \text{o} & & \text{x} \\ \hline \text{o} & \text{x} & \\ \hline \text{x} & \text{o} & \text{x} \end{array}\right) = \text{true}$$

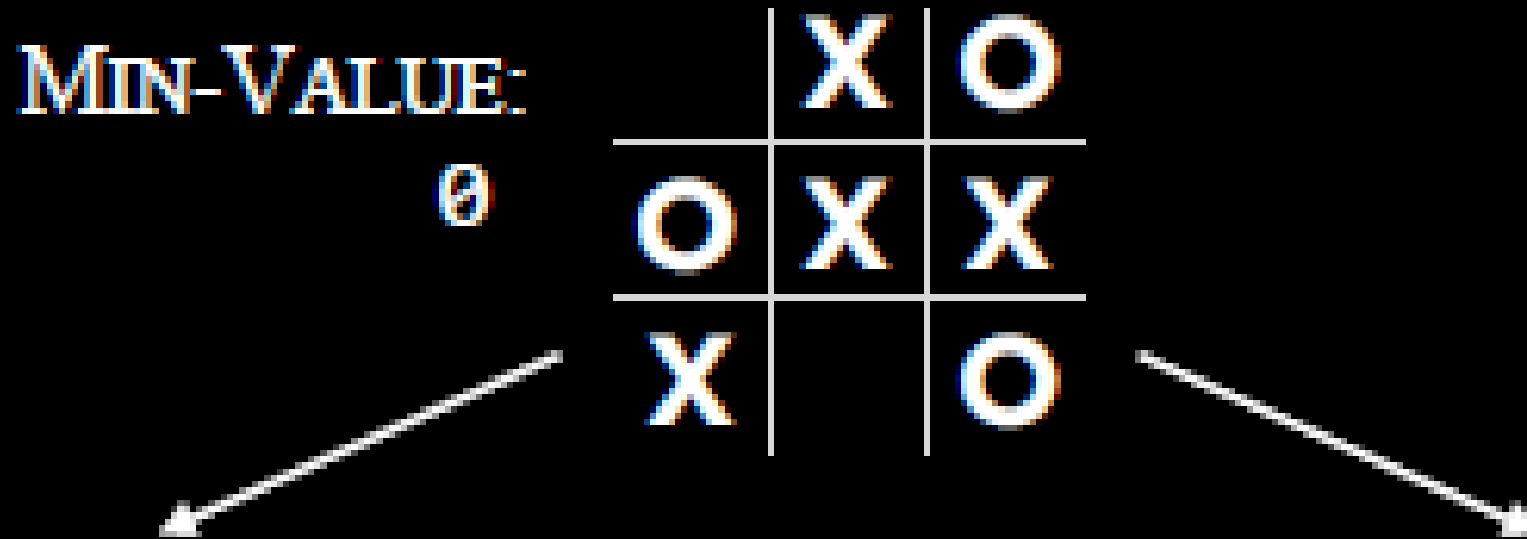
UTILITY(s)

$$\text{UTILITY}\left(\begin{array}{c|c|c} \text{O} & & \text{X} \\ \hline \text{O} & \text{X} & \\ \hline \text{X} & \text{O} & \text{X} \end{array} \right) = 1$$

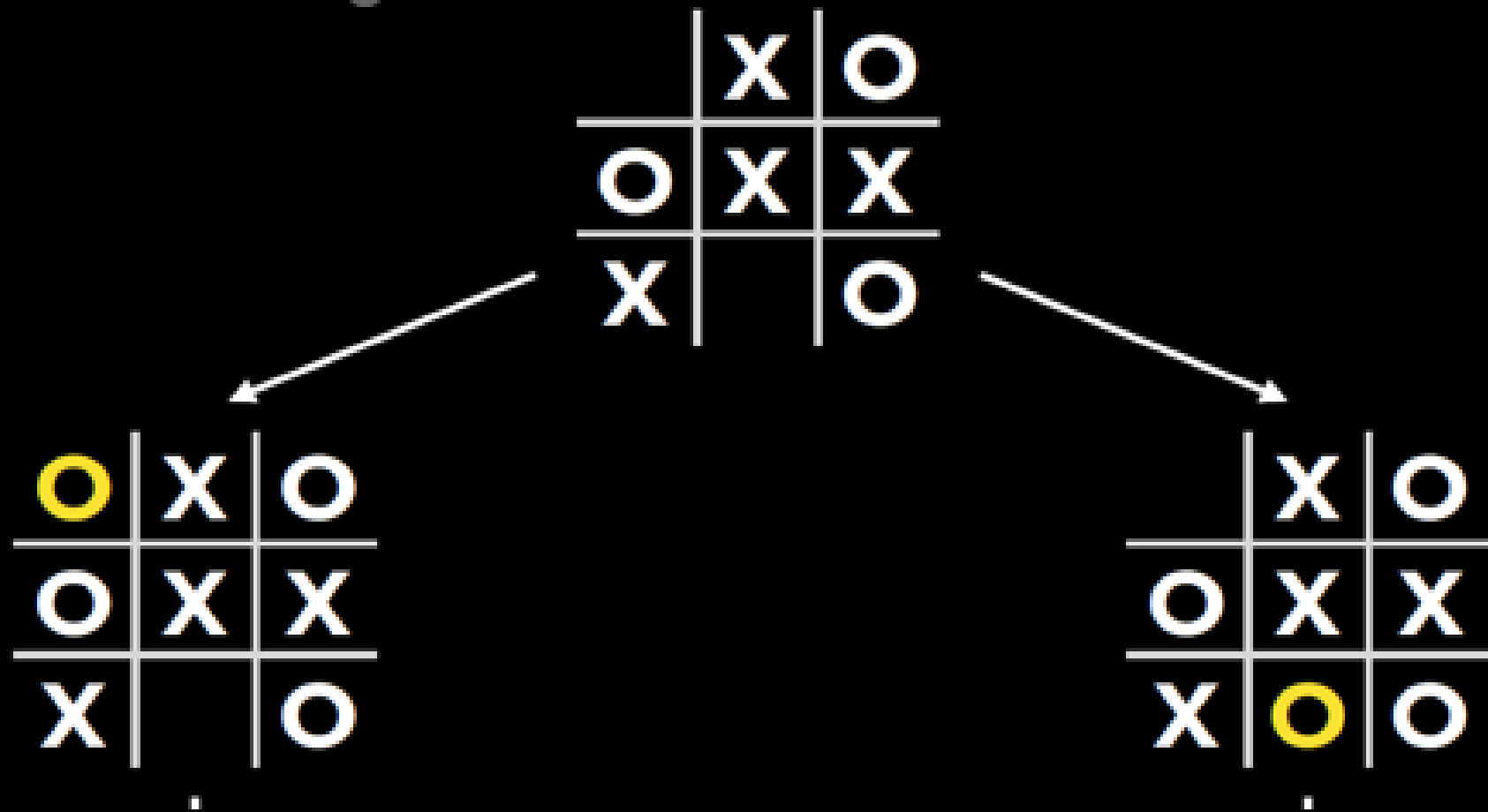
$$\text{UTILITY}\left(\begin{array}{c|c|c} \text{O} & \text{X} & \text{X} \\ \hline \text{X} & \text{O} & \\ \hline \text{O} & \text{X} & \text{O} \end{array} \right) = -1$$

FUNÇÃO UTILIDADE

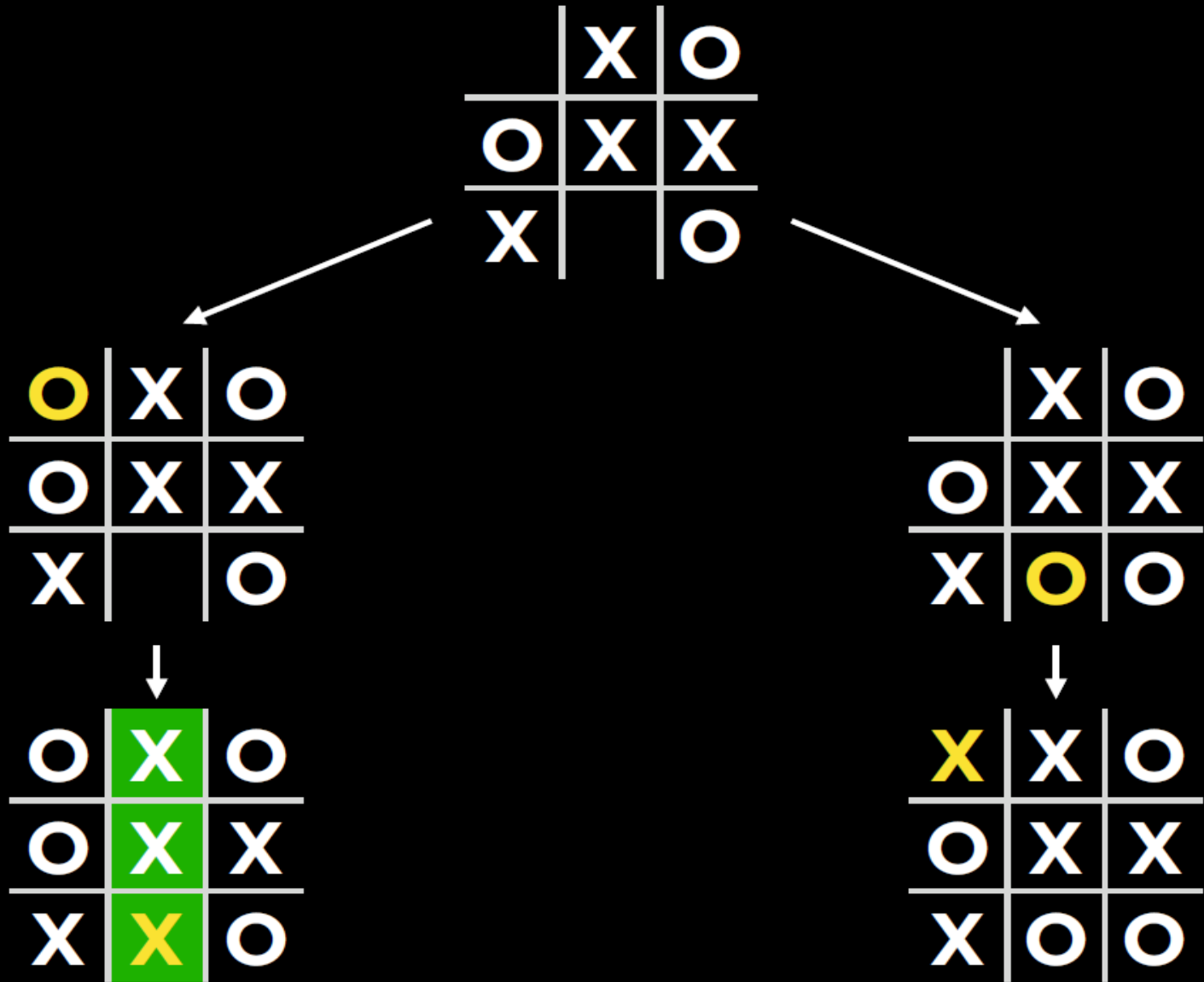
COMO FUNCIONA A FUNÇÃO UTILIDADE SE O JOGO AINDA NÃO TERMINOU?



DEVEMOS CONSIDERAR AS AÇÕES POSSÍVEIS E A QUAIS ESTADOS ELAS NOS LEVARÃO.

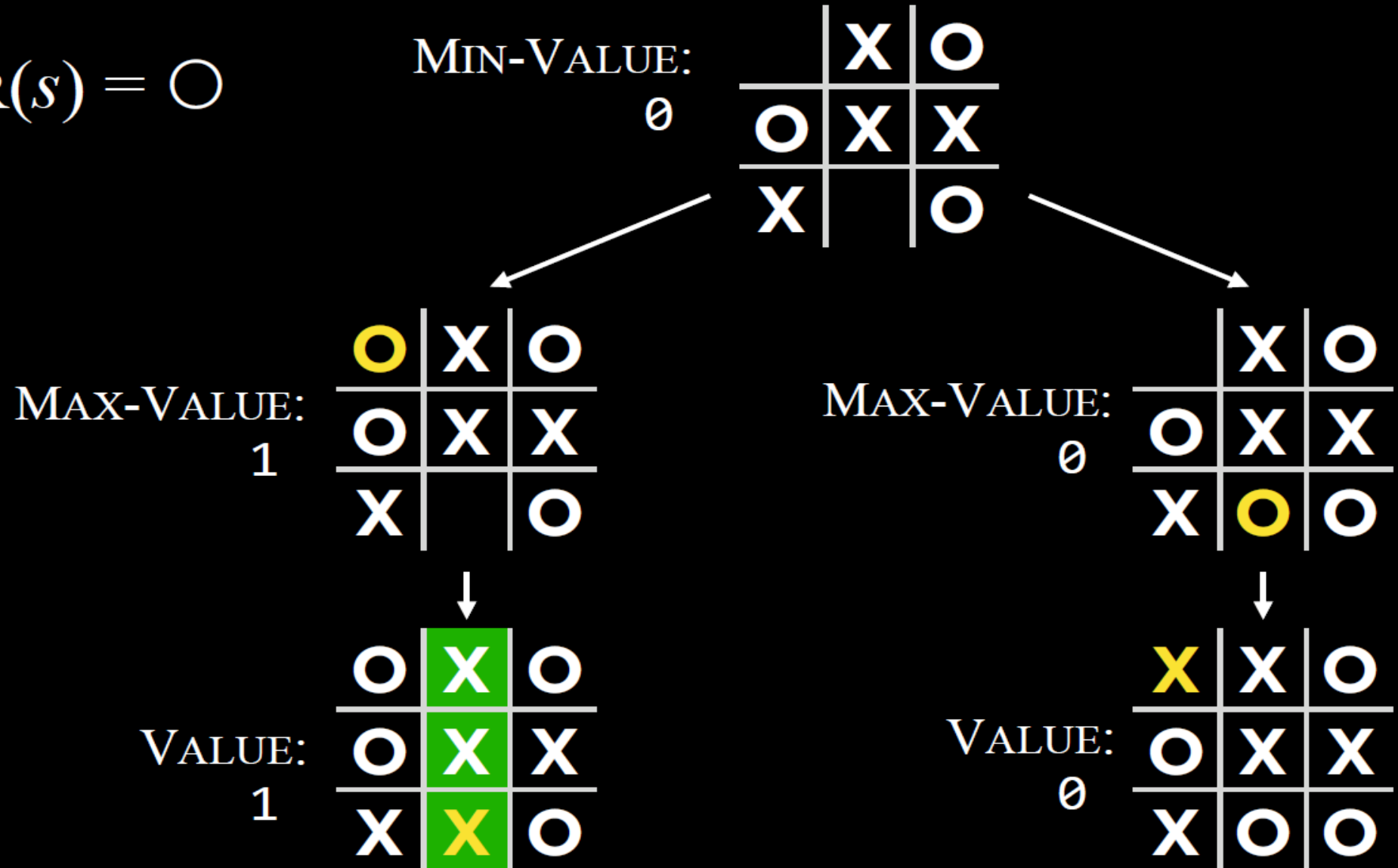


$$\text{PLAYER}(s) = \bigcirc$$



Jogador 'o' escolherá o caminho de menor valor.

PLAYER(s) = ○



PLAYER(s) = X

MAX-VALUE:
1

	X	O
O	X	
X		O

MIN-VALUE:
0

	X	O
O	X	X
X		O

MIN-VALUE:
-1

X	X	O
O	X	
X		O

VALUE:
1

X		O
O	X	
X	X	O

MAX-VALUE:
1

O	X	O
O	X	X
X		O

MAX-VALUE:
0

	X	O
O	X	X
X	O	O

VALUE:
-1

X	X	O
O	X	O
X		O

MAX-VALUE:
0

X	X	O
O	X	
X	O	O

VALUE:
1

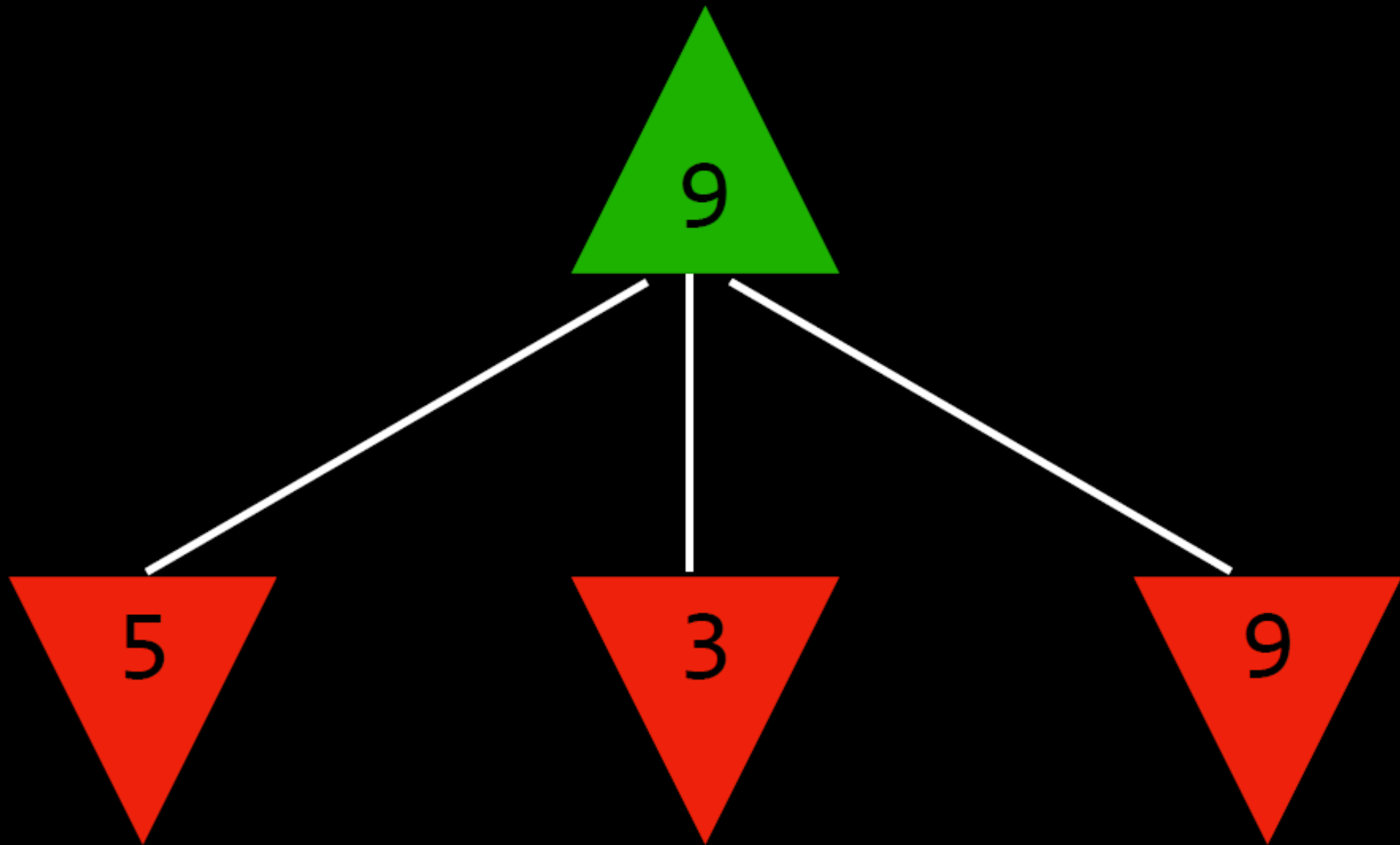
O	X	O
O	X	X
X	X	O

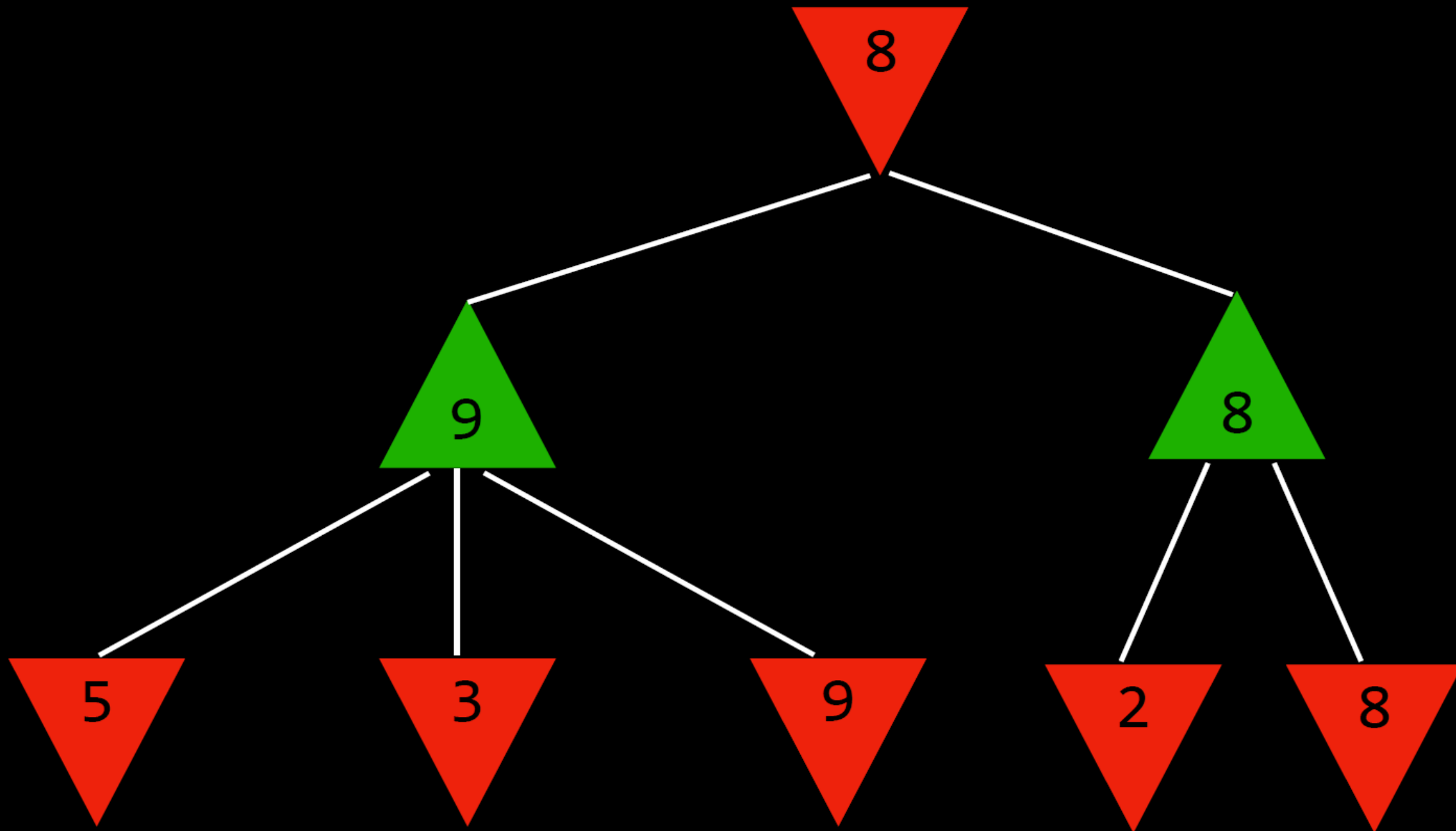
VALUE:
0

X	X	O
O	X	X
X	O	O

VALUE:
0

X	X	O
O	X	X
X	O	O





MiniMax

DADO UM ESTADO 'S':

**'MAX' ESCOLHE A AÇÃO 'A' DENTRO DE AÇÃO(S) QUE
PRODUZ O MAIOR VALOR PARA
MIN-VALUE (RESULTADO (S, A))**

**'MIN' ESCOLHE A AÇÃO 'A' DENTRO DE AÇÃO(S) QUE
PRODUZ O MENOR VALOR PARA
MAX-VALUE (RESULTADO (S, A)) .**

MiniMax

**COMO CALCULAR O VALOR DE UM ESTADO SE ESTAMOS
TENTANDO MAXIMIZAR OU MINIMIZAR SEU VALOR?**

Max

function MAX-VALUE (*state*):

 if TERMINAL (*state*):

return UTILITY (*state*)

v = -inf *#menor valor possível para o estado*

 for *action* in ACTIONS (*state*):

v = MAX (*v* , MIN-VALUE (RESULT (*state*, *action*)))

return v

Min

function MIN-VALUE (*state*):

 if TERMINAL (*state*):

return UTILITY (*state*)

$v = +\infty$

#maior valor possível para o estado

 for *action* in ACTIONS (*state*):

$v = \text{MIN} (v , \text{MAX-VALUE} (\text{RESULT} (\textit{state}, \textit{action})))$

return v

MiniMax

A APLICAÇÃO DOS ALGORITMOS Min \leftrightarrow Max
APRESENTADOS PERMITEM APLICAR AS FUNÇÕES MIN e
MAX RECURSIVAMENTE ATÉ SE ATINGIR UM ESTADO
TERMINAL.

OTIMIZAÇÃO

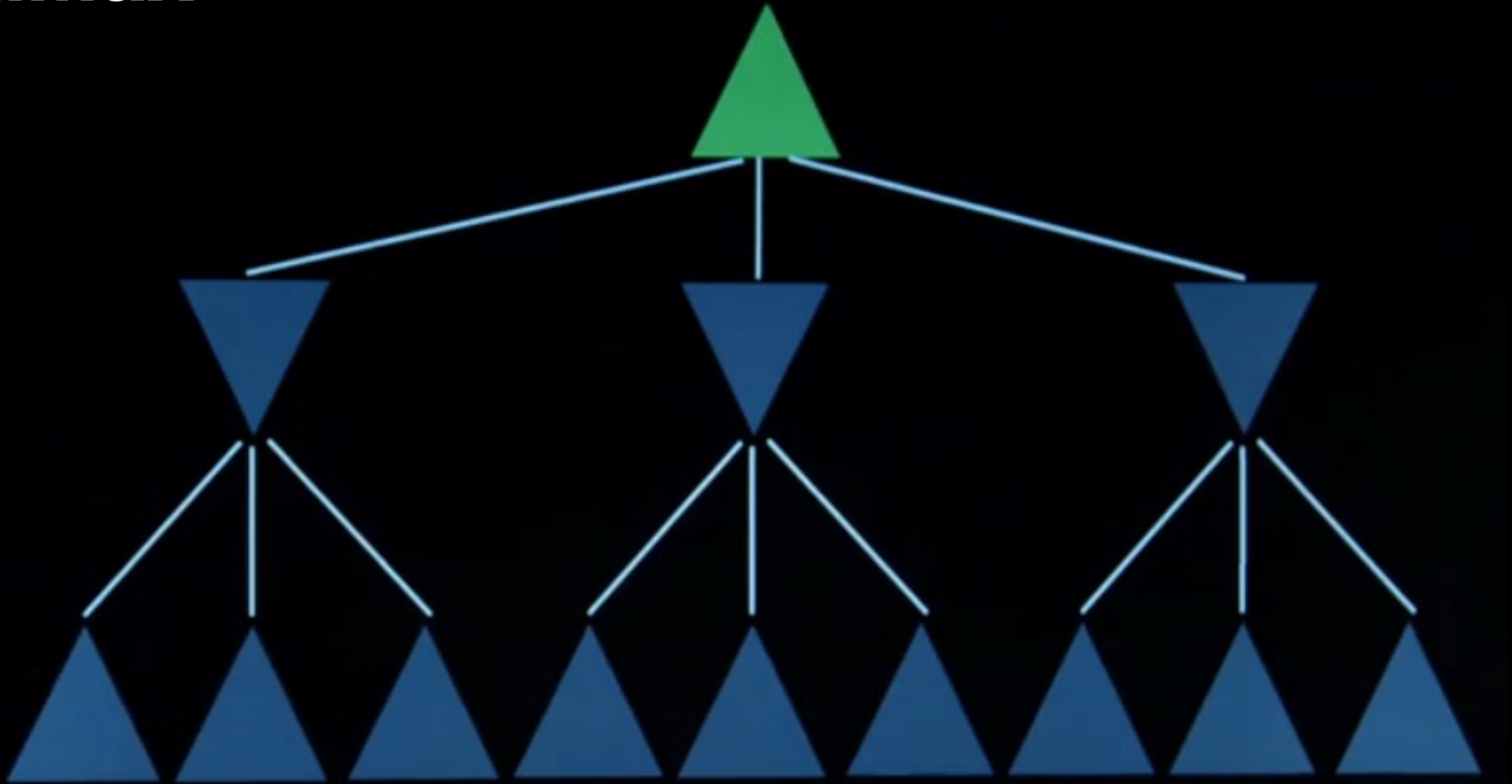
**Quantos são os estados possíveis do jogo
da velha?**

MiniMax

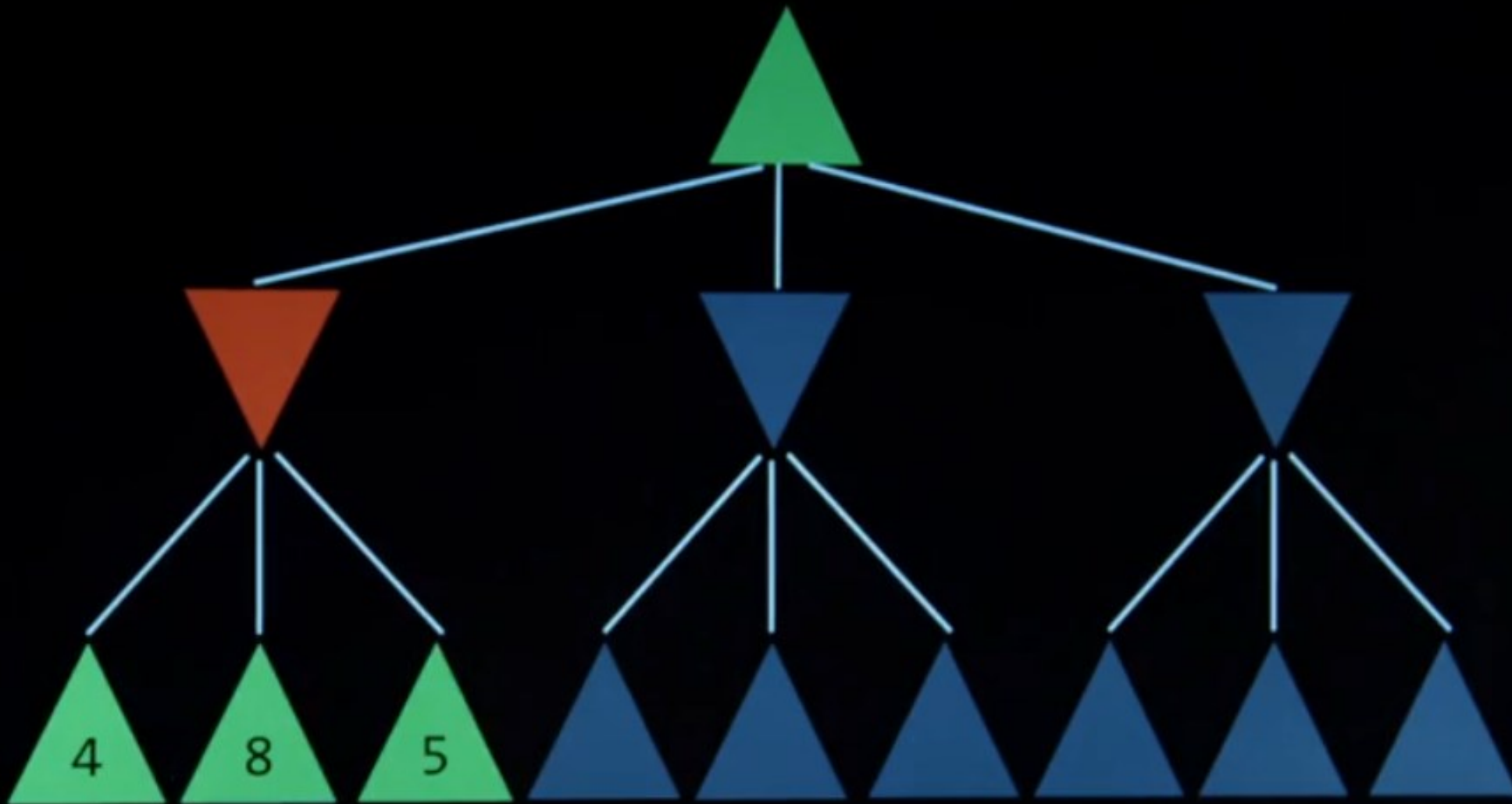
OTIMIZAÇÕES POSSÍVEIS ?

**Como gastar menos espaço ou tempo para
atingir o estado desejado?**

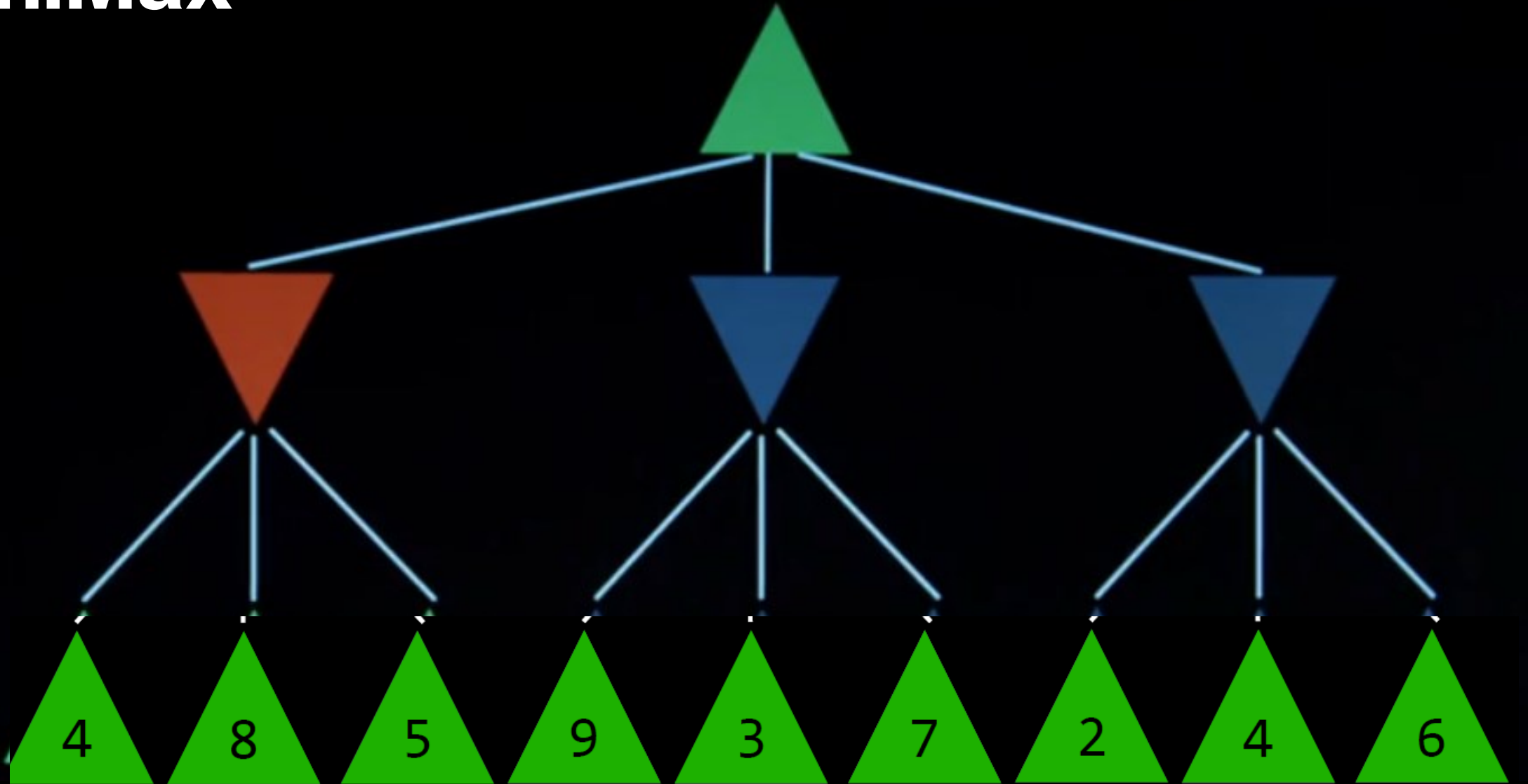
MiniMax



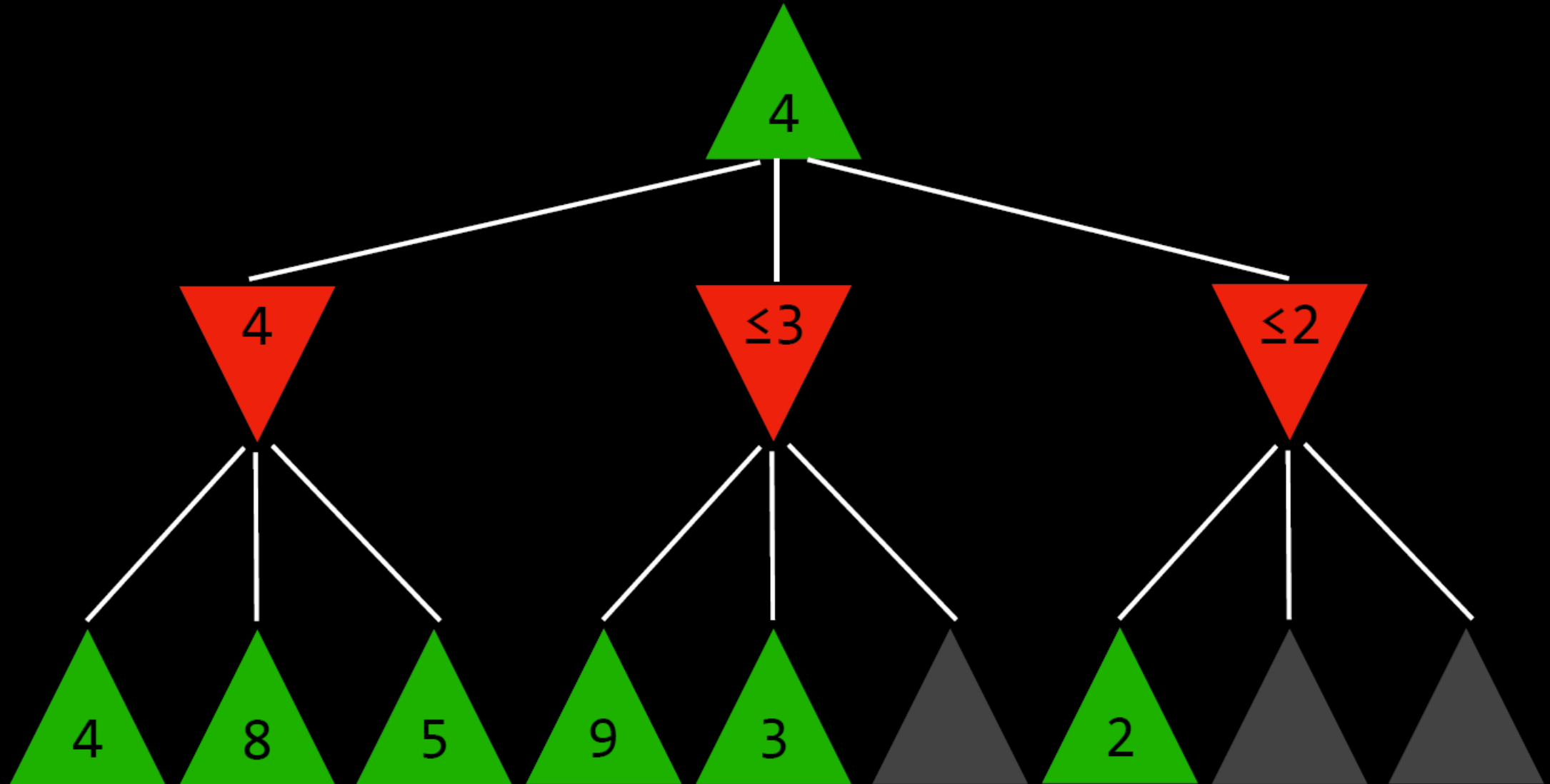
MiniMax



MiniMax



Alpha-Beta Pruning



OTIMIZAÇÃO

**Quantos são os estados possíveis do jogo
da velha?**

255.168

OTIMIZAÇÃO

Quantos são os estados possíveis do jogo de Xadrez após 4 movimentos de cada jogador?

288 Bi

OTIMIZAÇÃO

**Quantos são os estados possíveis do jogo de
Xadrez?**

Pelo menos 10^{29000}

Deep Limited MiniMax

- A previsão é encerrada após certo número rodadas, mesmo sem atingir um estado terminal.
- É necessário dar valor a um estado não-terminal do jogo.

EVALUATION FUNCTION

Retorna o valor estimado da função utilidade para um estado do jogo.

Quanto maior a precisão ao estimar o valor da função utilidade, melhor será a IA.

YOUR MOVE IS GIVEN BY THE POSITION OF THE LARGEST RED SYMBOL ON THE GRID. WHEN YOUR OPPONENT PICKS A MOVE, ZOOM IN ON THE REGION OF THE GRID WHERE THEY WENT. REPEAT.

MAP FOR X:

