

Programação Científica

Prof. Dr. Danilo H. Perico

JOGOS EM IA: ALGORITMO MINIMAX

JOGOS EM IA

- São domínios clássicos em **IA**
 - Abstratos: fáceis de formalizar e representar
 - Podem ter sua complexidade reduzida ou aumentada
 - Exigem a tomada de decisões (*muitas vezes em um curto intervalo de tempo*)
 - Há interação com o usuário

JOGOS EM IA

- Em **Teoria dos Jogos**, qualquer ambiente multiagente pode ser visto como um jogo, desde que cada agente possa exercer um impacto significativo nos outros agentes
- Em **IA**, jogos são, normalmente, determinísticos e do tipo **soma zero** com **informações perfeitas**:
 - Dois agentes cujas ações devem se alternar e que os valores de **utilidade** no final do jogo são sempre iguais e opostos
 - Exemplo: se um jogador ganha um jogo de xadrez (+1) o outro jogador, necessariamente, perde (-1)

JOGOS EM IA

- **Utilidade:** quando um estado é preferido em detrimento de outro, esse estado tem maior **utilidade** ao agente
- Assim, uma **função de utilidade** mapeia um estado (*ou uma sequência de estados*) em números reais, que descreve, basicamente, se um estado é melhor do que o outro para aquele problema

DECISÕES ÓTIMAS EM JOGOS

- A presença de um oponente introduz **incerteza** ao problema
 - O número de possíveis soluções passa a ser muito maior, pois o movimento do oponente é “imprevisível”
- É muito comum que não exista tempo suficiente para testar todas as possíveis consequências de um movimento durante um jogo

DECISÕES ÓTIMAS EM JOGOS

- Como, então, encontrar o melhor movimento?
- Existem algumas estratégias, uma delas é o **Algoritmo MINIMAX (MINMAX)**

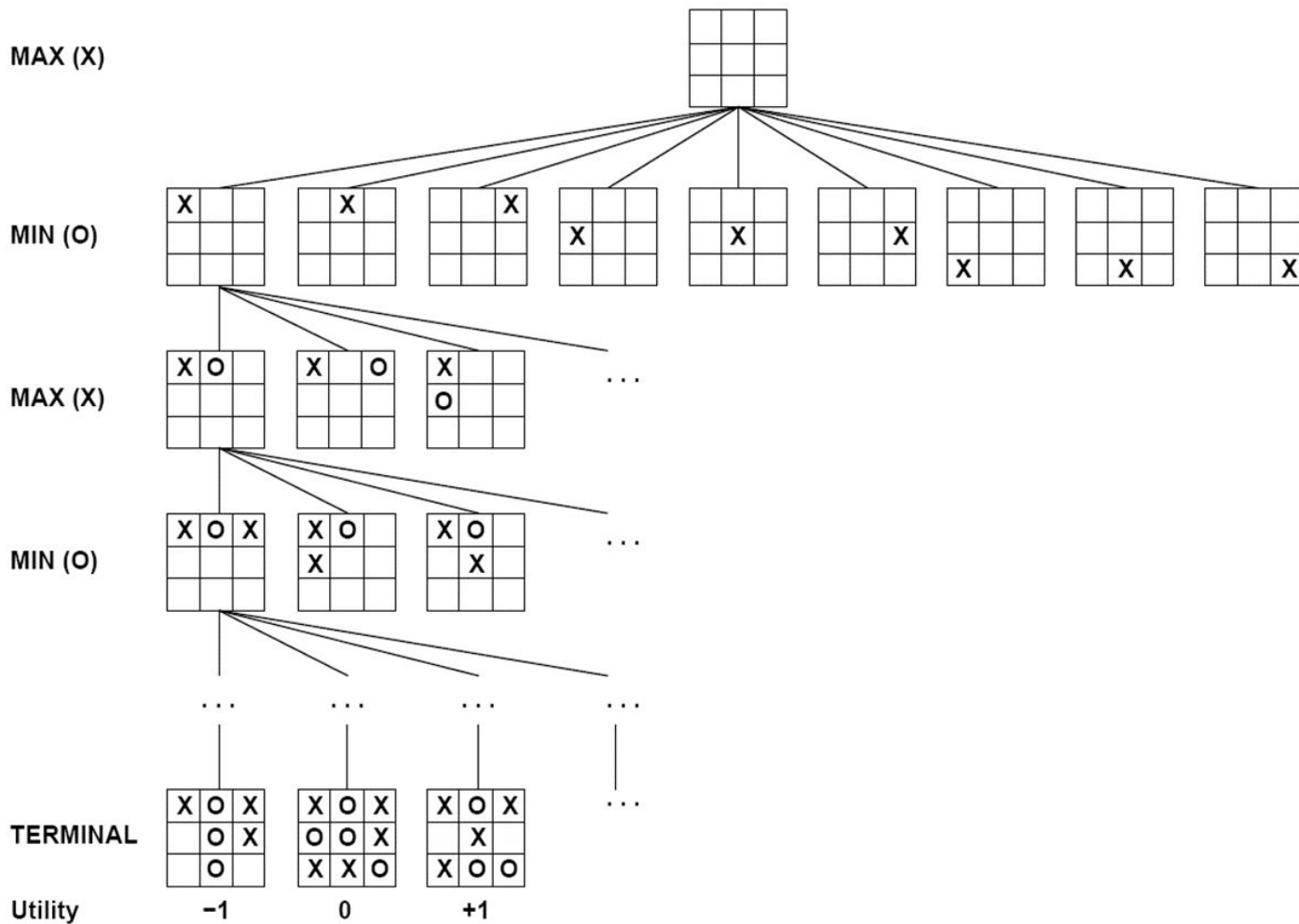
ALGORITMO MINIMAX

- **MINI**mizar a jogada do adversário
- **MAX**imizar a sua própria jogada
- Características:
 - Precisa da função de utilidade
 - Busca escolher movimentos que maximizem a sua chance e diminuam a chance do adversário
 - Busca sempre a melhor opção para o adversário na jogada dele

DECISÕES ÓTIMAS EM JOGOS

- Inicialmente jogos com dois jogadores:
 - **MAX** e **MIN**
 - **MAX** inicia o movimento, depois eles revezam
- Um jogo pode ser definido como uma **árvore (árvore do jogo)**:
 - Estado inicial
 - Função sucessor (movimento, estado)
 - Teste de término
 - Função de utilidade: dá um valor numérico para os estados terminais

EXEMPLO JOGO DA VELHA



DECISÕES ÓTIMAS EM JOGOS

- A solução ótima para **MAX** depende dos movimentos de **MIN**, logo:
 - **MAX** deve encontrar uma estratégia de contingência que especifique o movimento de **MAX** no estado inicial, e depois o movimento de **MAX** nos estados resultantes de cada movimento de **MIN** e assim por diante

DECISÕES ÓTIMAS EM JOGOS

- Dada uma árvore de jogo, a estratégia ótima pode ser determinada a partir do valor minimax de cada nó
- O valor minimax (para MAX) é a utilidade de MAX para cada estado, assumindo que MIN escolhe os estados mais vantajosos para ele mesmo (i.e. os estados com menor valor utilidade para MAX)

EXEMPLO JOGO DA VELHA

X ganha: **+1**

O ganha: **-1**

Empate: **0**

X		
X	O	O
O	X	

0

max

X	X	
X	O	O
O	X	

-1

X		X
X	O	O
O	X	

0

X		
X	O	O
O	X	X

-1

min

X	X	O
X	O	O
O	X	

-1

X	X	
X	O	O
O	X	O

+1

X	O	X
X	O	O
O	X	

0

X		X
X	O	O
O	X	O

+1

X	O	
X	O	O
O	X	X

0

X		O
X	O	O
O	X	X

-1

max

X	X	X
X	O	O
O	X	O

+1

X	O	X
X	O	O
O	X	X

0

X	X	X
X	O	O
O	X	O

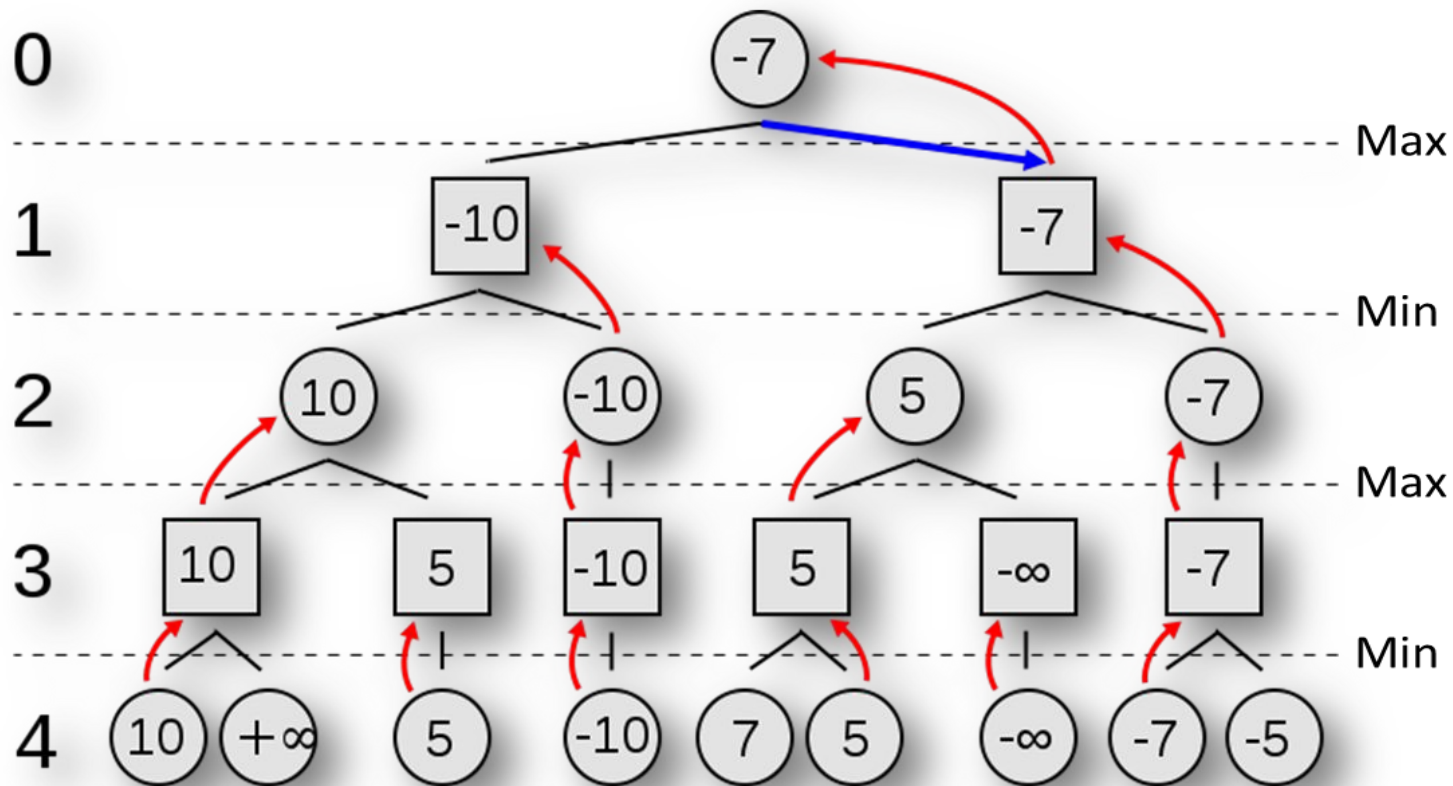
+1

X	O	X
X	O	O
O	X	X

0

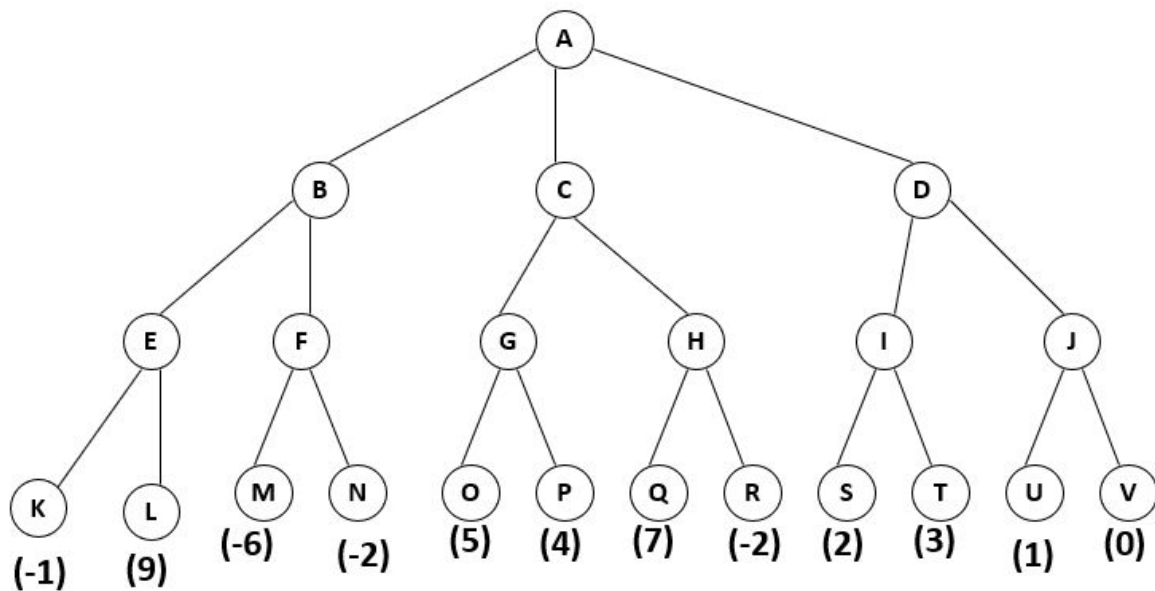
min

ESQUEMA GERAL DO MINIMAX



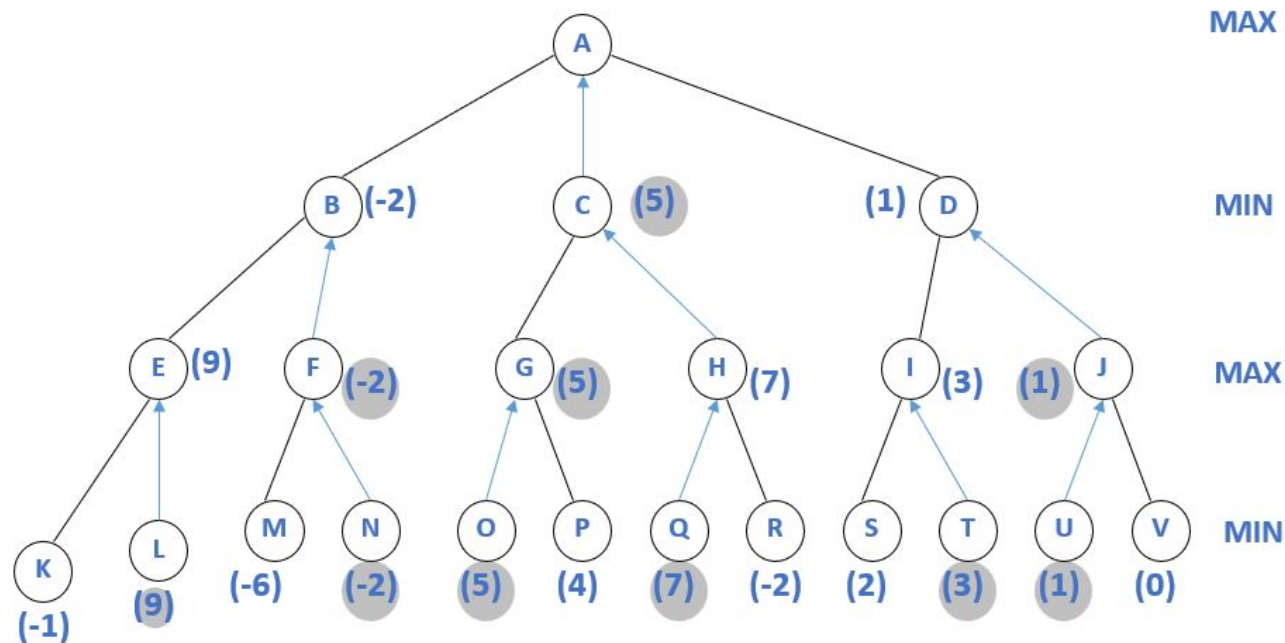
MINIMAX

- Qual a melhor jogada, B, C ou D ?



MINIMAX

- Qual a melhor jogada, B, C ou D ?

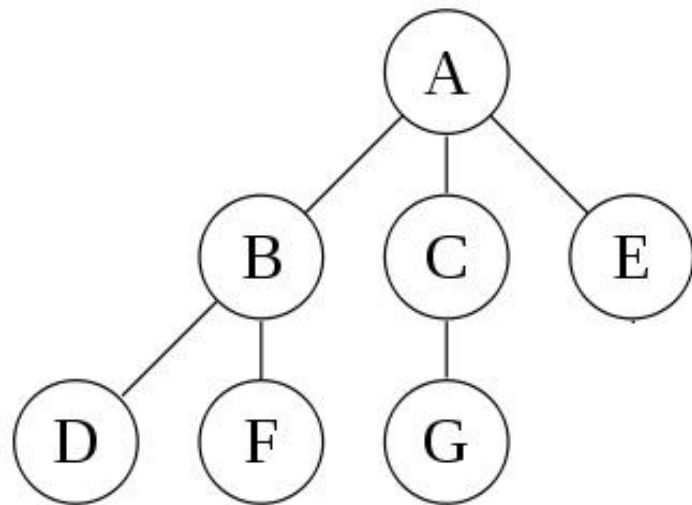


ALGORITMO MINIMAX

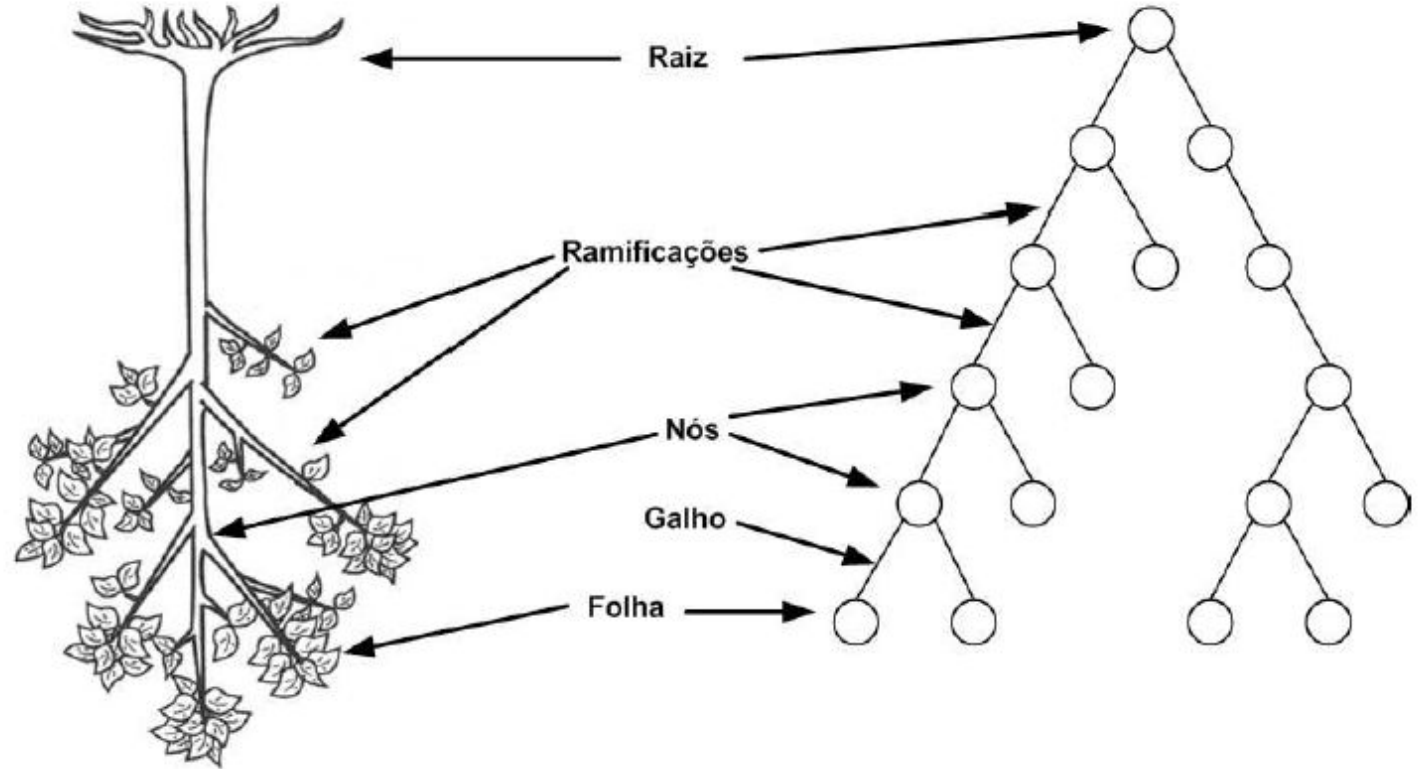
- Baseado na estrutura de **árvore**:
 - Cada nó representa o estado do jogo e contém a coleção de nós filhos que representam todos os movimentos a partir daquele estado
- Portanto, normalmente implementado com **recursividade**
 - Calcula recursivamente os valores de utilidade de cada estado

ÁRVORES

- Uma das mais importantes **estruturas de dados** não lineares
- Diferente das listas, em que os dados se encontram numa sequência, nas árvores **os dados estão dispostos de forma hierárquica**
- São **estruturas eficientes** e simples em relação ao tratamento computacional

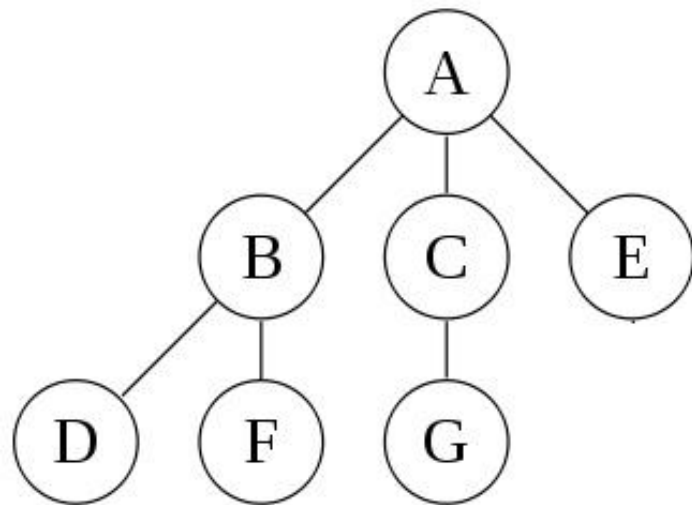


ÁRVORES



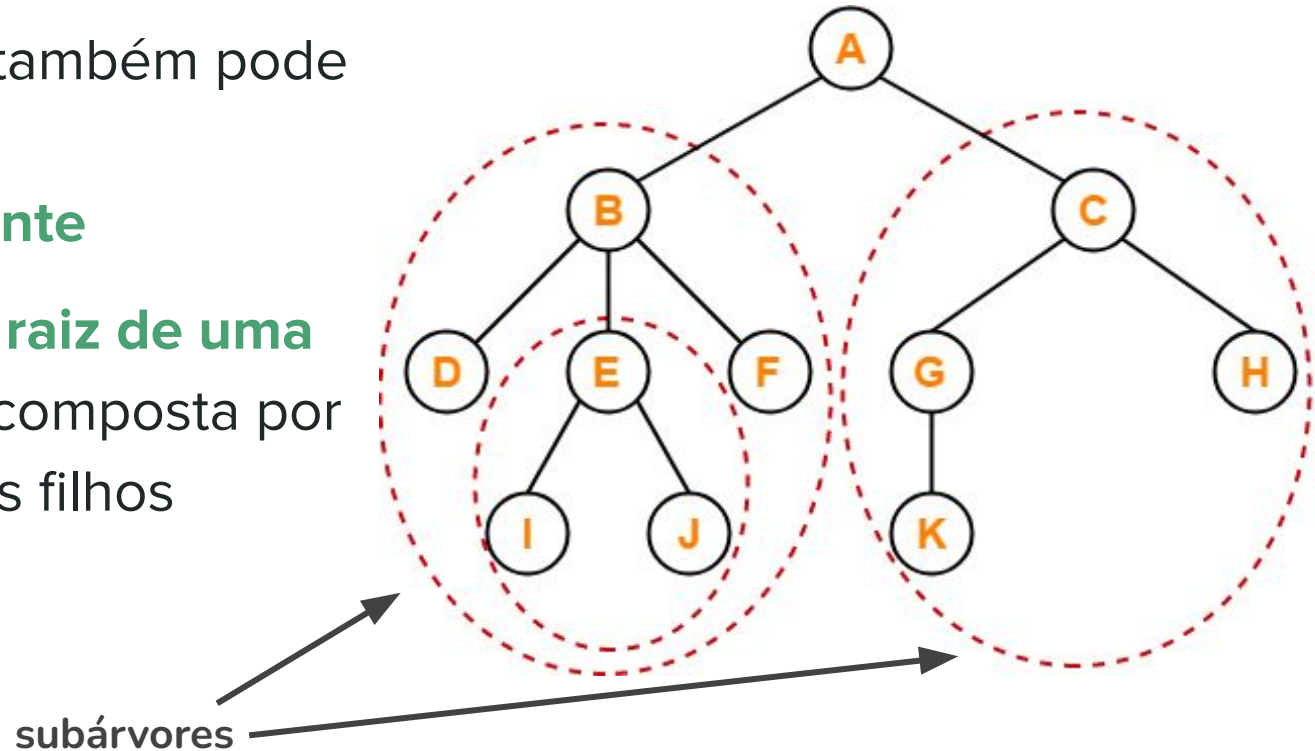
ÁRVORES

- Uma árvore é formada por um conjunto de **elementos que armazenam informações (nós)**
- Toda a árvore possui o elemento chamado **raiz**, que possui ligações para outros elementos denominados **ramos ou filhos**



ÁRVORES - DEFINIÇÃO RECURSIVA

- Uma árvore também pode ser definida **recursivamente**
- **Cada nó é a raiz de uma sub-árvore**, composta por seus próprios filhos



RECURSIVIDADE

- Recursividade é um termo usado de maneira mais geral para descrever o **processo de repetição** de um objeto de um jeito similar ao que já fora mostrado
- Um bom exemplo disso são as imagens repetidas que aparecem quando **dois espelhos são apontados um para o outro**

Recursividade



RECURSIVIDADE

- Em termos gerais, a recursão pode ser considerada como um processo de **repetição de um procedimento ou função**
- Portanto, de maneira bem simplista, pode ser definida como um **procedimento ou função que chama a si mesmo(a)**

ENTREGA

ENTREGA

- Aplicar o MINIMAX para o jogo da velha 3x3x3 (*jogo da velha em 3 dimensões*)
- A ideia é fazer o humano jogar contra a IA (*implementada com o MINIMAX*)
- Demonstração do jogo 4x4x4 (*com regras do jogo*):
- <https://www.mathsisfun.com/games/four-sight-3d-tic-tac-toe.html>

