

IA

Nome: \_\_\_\_\_

RA: \_\_\_\_\_

**Lista 04**

Professor: Vinicius Pereira

---

Esta lista contém 8 páginas e 5 questões.

---

**Procedimento minimax :**

Se a árvore é exaustivamente buscável abra todos os vértices e escolha uma avaliação para cada resultado.

Se não, abra até certo nível e escolha uma heurística de avaliação para cada vértice resultado.

Na implementação, rotulamos cada nível da busca de acordo com quem joga neste ponto do jogo: MIN, ou MAX.

Cada vértice folha recebe uma avaliação.

O procedimento propaga estes valor para os vértices ancestrais de acordo com a seguinte regra:

Se o estado pai for um vértice MAX dê-lhe o valor máximo entre seus filhos - esta é a escolha do jogador que quer maximizar o ganho.

Se o estado pai for um vértice MIN dê-lhe o valor mínimo entre seus filhos - esta é a escolha do jogador que quer minimizar o ganho do jogador MAX.

**Poda Alfa-Beta** O procedimento inicial é fazer uma busca em profundidade para termos um valor inicial, com este valor criamos valores  $\alpha$  e  $\beta$  para os vértices ancestrais da folha encontrada.

Associa um valor  $\alpha$  para cada vértice de MAX, este valor representa o melhor (mais alto) valor que foi encontrado até agora nos caminhos de uma folha para este vértice.

Associa um valor  $\beta$  para cada vértice de MIN, este valor representa o melhor (mais baixo) valor que foi encontrado até agora nos caminhos de uma folha para este vértice.

A seguinte regra é aplicada:

A poda pode ser feita abaixo de qualquer vértice MIN que tenha um valor  $\beta$  menor, ou igual, que o valor  $\alpha$  de qualquer um dos seus ancestrais MAX.

A poda pode ser feita abaixo de qualquer vértice MAX que tenha um valor  $\alpha$  maior, ou igual, que o valor  $\beta$  de qualquer um dos seus ancestrais MIN.

Note que valores  $\alpha$  não diminuem e valores  $\beta$  não aumentam.

1. Considere a árvore da figura 1, propague os valores das folhas até a raiz, usando o procedimento Minimax.

Faça a poda alfabeta da esquerda para a direita e da direita para a esquerda.

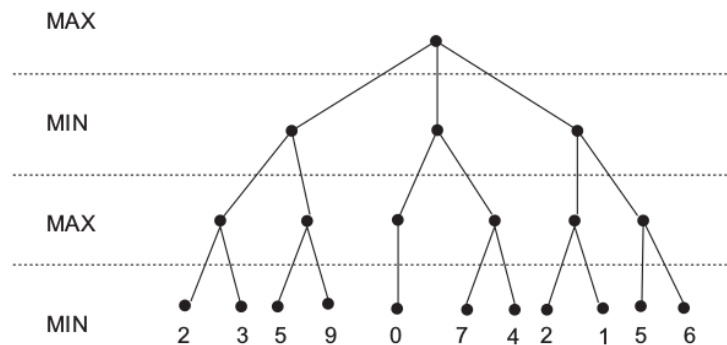


Figura 1: Uma árvore para o procedimento Minimax.

2. Considere a árvore da figura 2, propague os valores das folhas até a raiz, usando o procedimento Minimax.

Faça a poda alfabetica da esquerda para a direita e da direita para a esquerda.

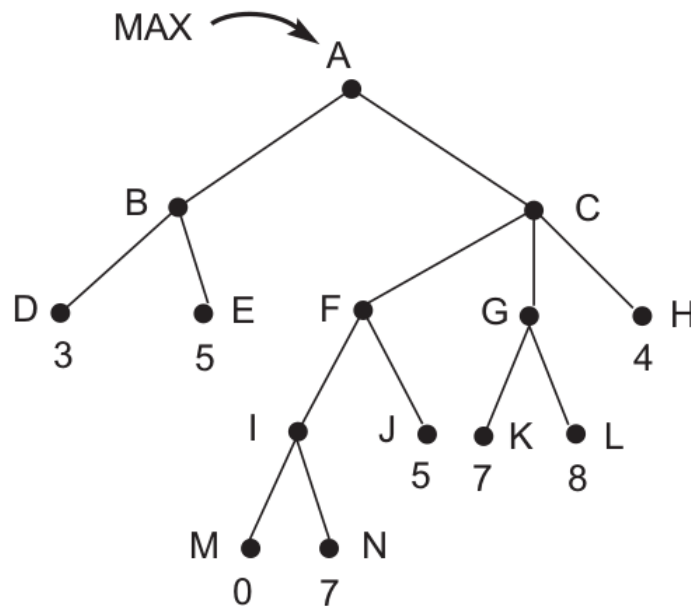


Figura 2: Uma árvore para o procedimento Minimax.

3. Considere uma variante do jogo *nim*, diversas fichas são colocadas sobre a mesa formando pilhas, em cada rodada um dos jogadores deve dividir uma das pilhas em duas pilhas sem que as duas pilhas resultantes tenham quantidade de fichas iguais. O jogador que não conseguir fazer uma jogada perde o jogo.

Para o jogo com dois jogadores, com o jogador MAX começando, faça uma busca exaustiva (até o último nível) para um estado inicial de uma pilha com 8 fichas, e para um estado inicial de uma pilha com 9 fichas. Observe estados semelhantes.

Atribua um valor para cada folha e faça a propagação dos valores das folhas nas duas árvores, evidenciando a jogada de cada jogador.

4. Considere o seguinte jogo “simples” para duas pessoas. Em um tabuleiro com  $n$  casas alinhadas, enumeradas de 1 a  $n$ , sendo a casa  $i$  vizinha à casa  $i + 1$  e  $i - 1$  e a mais nenhuma outra, a casa 1 vizinha à casa 2 e a casa  $n$  vizinha à casa  $n - 1$  e a mais nenhuma outra. Cada jogador tem apenas uma peça que pode ocupar uma casa. Em cada turno um jogador deve mover a sua peça e este movimento pode ser para uma casa vizinha, caso esta esteja vaga; ou pular a peça do adversário movendo-se para a casa vizinha à peça do adversário, caso a peça do adversário esteja em uma casa vizinha. A peça do jogador  $A$  começa na casa 1 e a peça do jogador  $B$  começa na casa  $n$ . O objetivo de cada jogador é chegar com a sua peça na casa outra extremidade, quem chegar primeiro ganha.

Um exemplo deste jogo com 4 casas pode ser visto na figura 3.

Abra uma árvore de estados para o jogo com 4 e com 5 casas. Reprunte cada estado como  $(S_A, S_B)$ , onde  $S_X$  é o número da casa do jogador  $X$ . Caso algum caminho entre em loop, junte os dois vértices repetidos e coloque seu valor como ‘?’.

Qual jogador ganhará em cada caso?

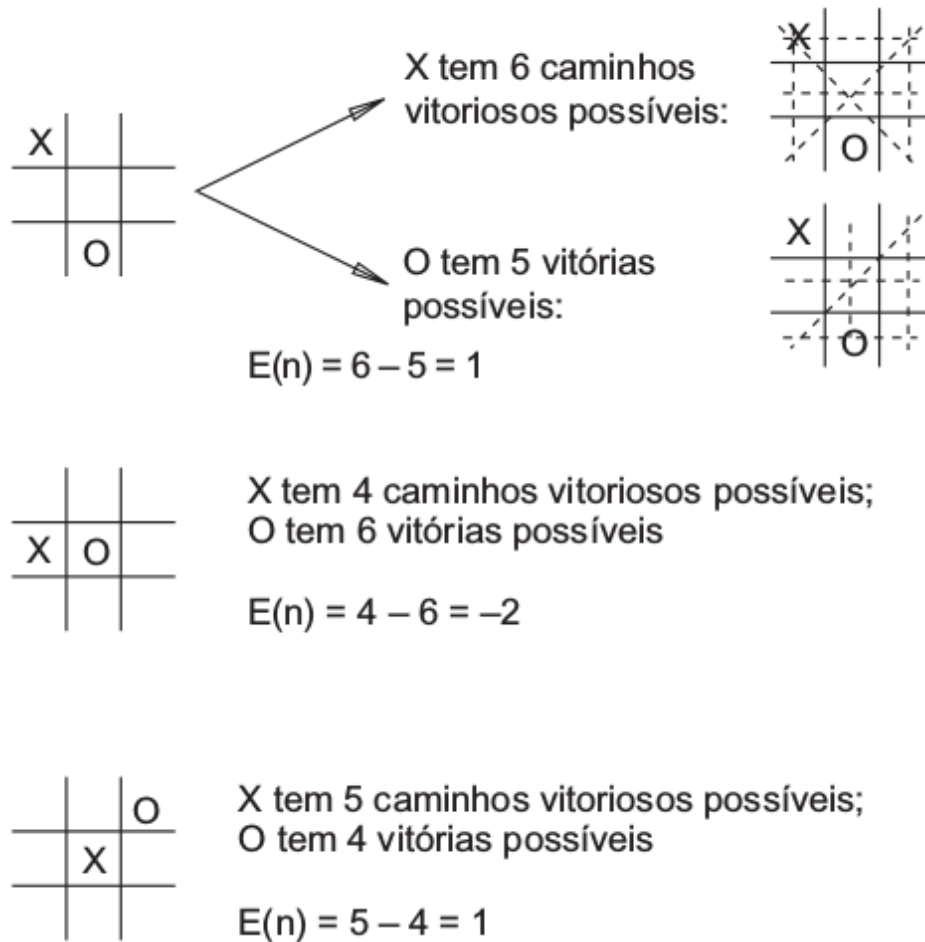


Figura 3: O jogo “simples” com 4 casas.

5. Considere a heurística da figura 4 para o jogo da velha.

As árvores das figuras 5, 6 e 7 são execuções do algoritmo Minimax com dois níveis de profundidade.

Redesenhe estas árvores aplicando a poda alfabeta, sendo que você pode escolher a ordem dos caminhos a percorrer.



A heurística é  $E(n) = M(n) - O(n)$

onde  $M(n)$  é o total de minhas linhas vitoriosas possíveis

$O(n)$  é o total de linhas vitoriosas possíveis do oponente

$E(n)$  é a avaliação total para o estado  $n$

Figura 4: Heurística para o jogo da velha.

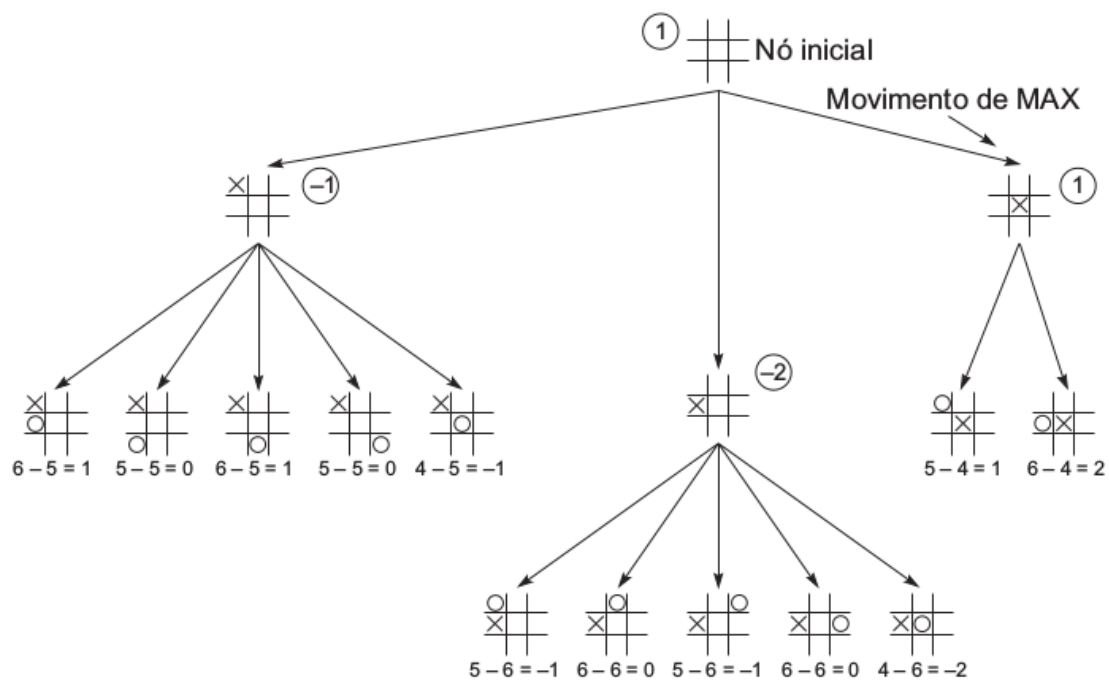


Figura 5: Iteração 1 para o procedimento Minimax com dois níveis.

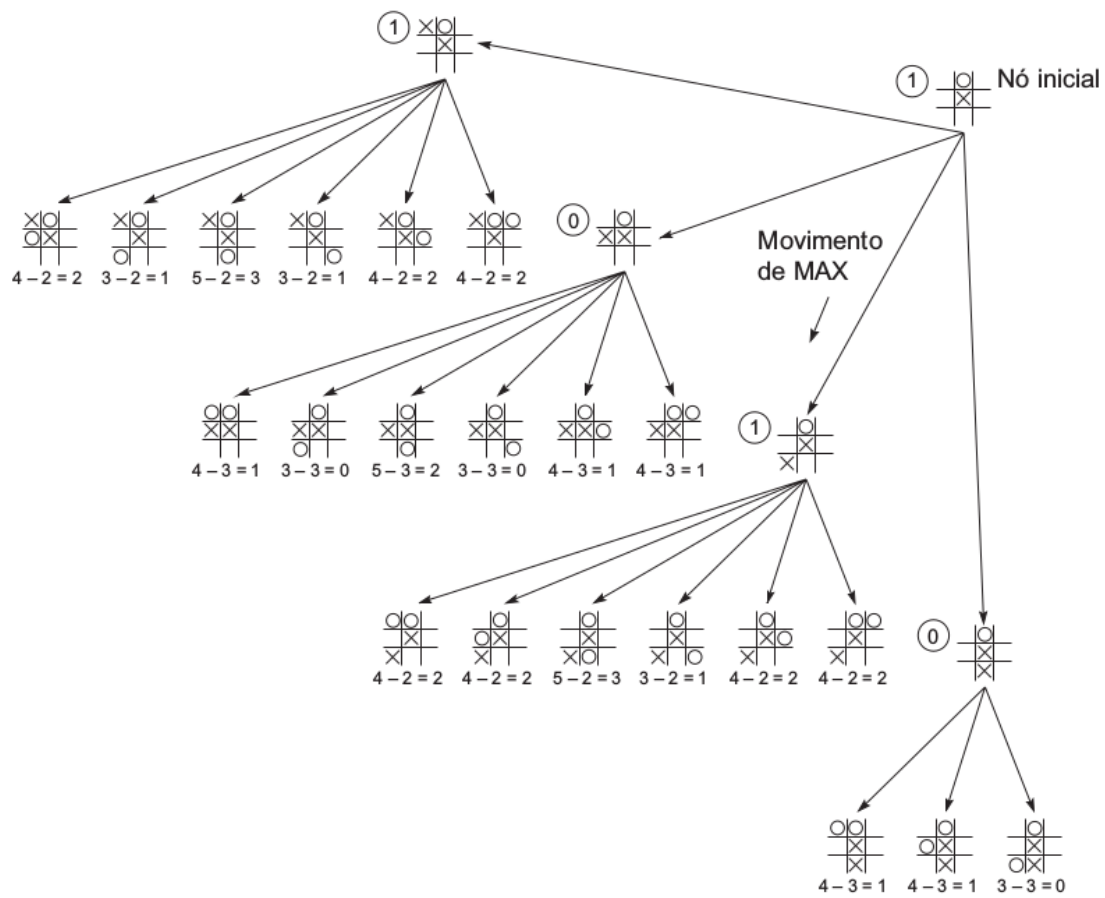


Figura 6: Iteração 2 para o procedimento Minimax com dois níveis.

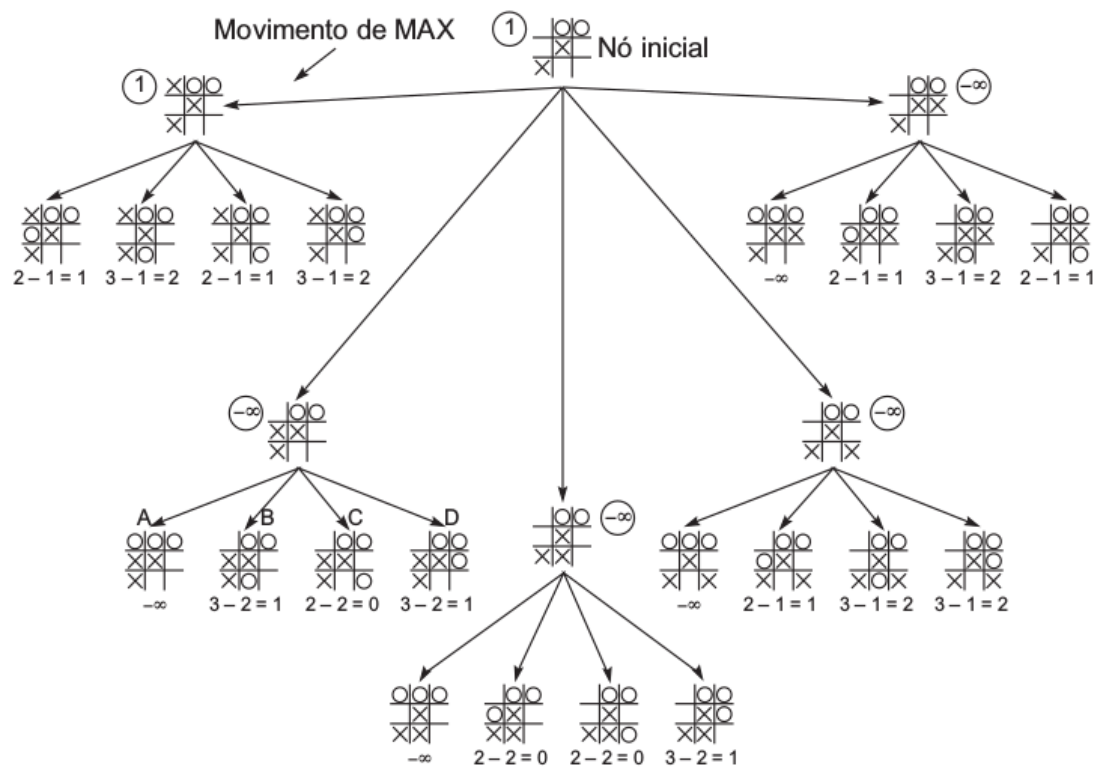


Figura 7: Iteração 3 para o procedimento Minimax com dois níveis.