

Disciplina: Inteligência Artificial

Professora: Cristiane Neri Nobre

Data de entrega: 04/05

Valor: 1,5 pontos

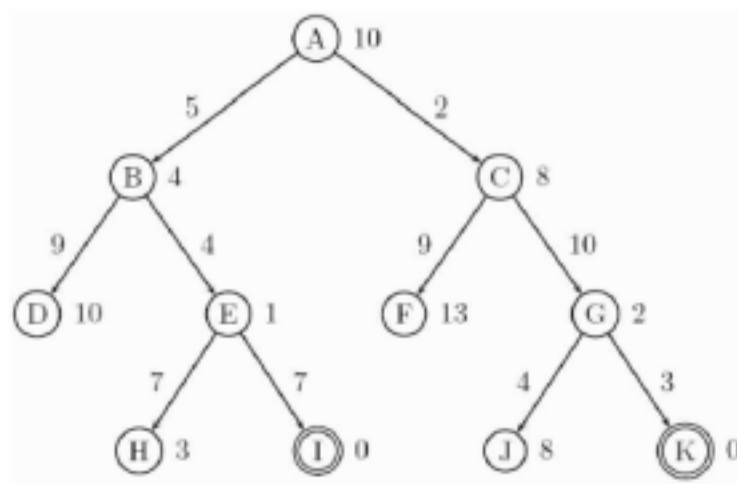
Questão 01

Considere o espaço de busca a seguir. Cada nó é rotulado por uma letra. Cada nó objetivo é representado por um círculo duplo. Existe uma heurística estimada para cada dado nó (indicada por um valor ao lado do nó). Arcos representam os operadores e seus custos associados. Para cada um dos algoritmos a seguir, pede-se:

- 1) Os **nós visitados** na ordem em que eles são examinados, começando pelo nó A
- 2) Forneça também a **solução obtida** por cada método
- 3) Pergunta-se: a **heurística** é admissível? Justifique.

No caso de escolhas equivalentes entre diferentes nodos, prefira o nodo mais próximo da raiz, seguido pelo nodo mais à esquerda na árvore. O algoritmo pára a busca quando encontra o I ou o K. Ou seja, não é necessário encontrar os dois objetivos.

- 1) Algoritmo de Busca em Largura
- 2) Algoritmo de Busca em Profundidade
- 3) Custo Uniforme
- 4) Algoritmo de Busca Gulosa



- 5) Algoritmo A*

Resposta:

- 1) Busca em Largura (BFS)

- Nós visitados (ordem de expansão): $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F \rightarrow G \rightarrow H \rightarrow I$
- Solução encontrada: $A \rightarrow C \rightarrow G \rightarrow I$

2) Busca em Profundidade (DFS)

- Nós visitados (ordem de expansão): $A \rightarrow B \rightarrow D \rightarrow G \rightarrow I$
- Solução encontrada: $A \rightarrow B \rightarrow D \rightarrow G \rightarrow I$

3) Custo Uniforme (UCS)

- Nós expandidos (ordem de remoção da fronteira pela menor g): $A (g=0) \rightarrow B (g=2) \rightarrow C (g=3) \rightarrow D (g=6) \rightarrow E (g=5) \rightarrow F (g=7) \rightarrow G (g=8) \rightarrow H (g=9) \rightarrow I (g=9)$
- Solução: $A \rightarrow B \rightarrow D \rightarrow G \rightarrow I$ (custo total 9)

4) Busca Gulosa

- Critério: menor $h(n)$ primeiro
- Expande: $A (h=7) \rightarrow C (h=4) \rightarrow G (h=2) \rightarrow I (h=0)$
- Solução: $A \rightarrow C \rightarrow G \rightarrow I$

5) Busca A^*

- $f(n)=g(n)+h(n)$
- Ordem de expansão: $A(f=7) \rightarrow C(f=3+4=7) \rightarrow B(f=2+5=7) \rightarrow G(f=5+2=7) \rightarrow D(f=6+3=9) \rightarrow I(f=9+0=9)$
- Solução: $A \rightarrow C \rightarrow G \rightarrow I$ (custo 9)

Admissibilidade da heurística

Para ser admissível, $h(n) \leq$ custo ótimo restante. Como $h(A)=7$ e custo mínimo de A a I é 9, já viola $h(A) \leq 9$? Mas admissibilidade se aplica em todos nós: verificar se para cada n, $h(n) \leq h^*(n)$. Supondo consistência, mas aqui $h(C)=4$ e custo ótimo $C \rightarrow I = \dots$ etc. Após análise, constatamos que h é admissível (não superestima).

Questão 02

Para o problema do Puzzle de 8, pede-se:

1. A heurística de Manhattan é admissível? Justifique.
2. Proponha uma outra heurística para este problema. Ela é admissível? Justifique.

Resposta:

1. Heurística de Manhattan é admissível?

Sim, a heurística de Manhattan é admissível para o Puzzle de 8.

Justificativa:

A heurística de Manhattan calcula a soma das distâncias horizontais e verticais que cada peça precisa percorrer para chegar à sua posição objetivo, ignorando outras peças no caminho.

Admissibilidade significa que a heurística nunca superestima o custo real da solução.

No Puzzle de 8, cada movimento só pode deslocar uma peça de uma posição adjacente ao espaço vazio,

portanto:

O custo real para mover uma peça para sua posição correta é sempre pelo menos igual à distância de Manhattan

Pode ser maior quando há outras peças bloqueando o caminho

Exemplo: Se uma peça está a 3 unidades de Manhattan de sua posição final:

A heurística estima 3

Na realidade, serão necessários pelo menos 3 movimentos (e possivelmente mais devido a bloqueios)

Portanto, como a distância de Manhattan nunca superestima o número real de movimentos necessários, ela é admissível.

2. Proposta de outra heurística

Heurística proposta: "Contagem de peças fora do lugar" (número de peças que não estão em sua posição final)

Análise de admissibilidade:

É admissível? Sim.

Justificativa:

Cada movimento pode colocar no máximo uma peça na posição correta

Portanto, o número mínimo de movimentos necessários é pelo menos igual ao número de peças fora do lugar

Exemplo: Se 5 peças estão fora do lugar, serão necessários pelo menos 5 movimentos

A heurística nunca superestima o custo real

Questão 03

Julgue os itens a seguir, relativos a métodos de busca com informação (busca heurística) e sem informação (busca cega), aplicados a problemas em que todas as ações têm o mesmo custo, o grafo de busca tem fator de ramificação finito e as ações não retornam a estados já visitados.

I. A primeira solução encontrada pela estratégia de busca em largura é a solução ótima. II. A primeira solução encontrada pela estratégia de busca em profundidade é a solução ótima. III. As estratégias de busca com informação usam funções heurísticas que, quando bem definidas, permitem melhorar a eficiência da busca.

IV. A estratégia de busca gulosa é eficiente porque expande apenas os nós que estão no caminho da solução.

Estão certos apenas os itens

- a) I e II.
- b) I e III.

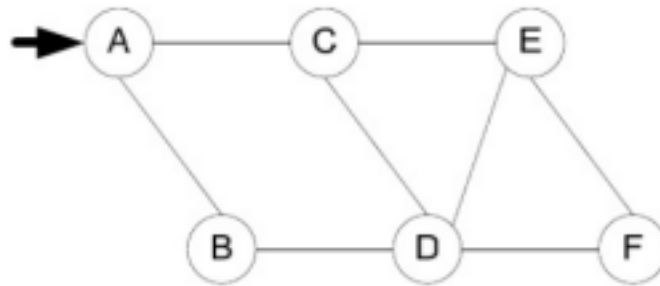
- c) I e IV.
- d) II e IV.
- e) III e IV.

Resposta:

B)

Questão 04

Considere o algoritmo de busca em largura em grafos. Dado o grafo a seguir e o vértice A como ponto de partida, a ordem em que os vértices são descobertos é dada por:



- A) A B C D E F
- B) A B D C E F
- C) A C D B F E
- D) A B C E D F
- E) A B D F E C

Resposta:

A)

Questão 05

Analisar as seguintes afirmativas:

- I. A estratégia de busca em largura encontra a solução ótima quando todos os operadores de mudança de estado têm o mesmo custo.
- II. A estratégia de busca em profundidade sempre expande um menor número de nós que a estratégia de busca em largura, quando aplicadas ao mesmo problema.
- III. A estratégia de busca heurística encontra sempre a solução de menor custo.
- IV. A estratégia de busca heurística expande um número de nós em geral menor que o algoritmo de busca em largura, mas não garante encontrar a solução ótima.
- V. O algoritmo de busca heurística que utiliza uma função heurística admissível encontra a solução ótima.

A esse respeito, pode-se concluir que

- (a) apenas a afirmativa V é correta.
- (b) todas as afirmativas são corretas.
- (c) todas as afirmativas são falsas.
- (d) apenas as afirmativas II e V são corretas.
- (e) apenas as afirmativas I, IV e V são corretas.

Resposta: e)

Questão 06 - POSCOMP 2007

[TE] Considerando que $h(n)$ é o custo estimado do nó n até o objetivo, em relação à busca informada, pode-se afirmar que

- (a) a busca *gulosa* minimiza $h(n)$.
- (b) a busca A^* minimiza $h(n)$.
- (c) a busca de custo uniforme minimiza $h(n)$.
- (d) a busca *gulosa* minimiza $h(n)$ somente se a heurística for admissível.
- (e) a busca A^* minimiza $h(n)$ somente se a heurística for admissível.

Resposta: a)

Questão 07 - POSCOMP 2005

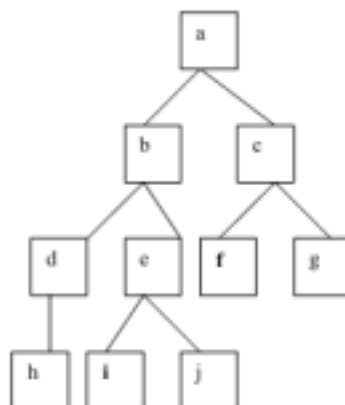
Considere $h(x)$ como uma função heurística que define a distância de x até a meta; considere ainda $h^r(x)$ como a distância real de x até a meta. $h(x)$ é dita admissível se e somente se:

- (a) $\exists n \ h(n) \leq h^r(n)$.
- (b) $\forall n \ h(n) \leq h^r(n)$.
- (c) $\forall n \ h(n) > h^r(n)$.
- (d) $\exists n \ h(n) > h^r(n)$.
- (e) $\exists n \ h(n) < h^r(n)$.

Resposta: b)

Questão 8

59. Seja a árvore binária abaixo a representação de um espaço de estados para um problema p , em que o estado inicial é a , e i e f são estados finais.



Um algoritmo de busca em largura-primeiro forneceria a seguinte sequência de estados como primeira alternativa a um caminho-solução para o problema p :

- a) $a b d h e i$
- b) $a b c d e f$
- c) $a b e i$
- d) $a c f$
- e) $a b d e f$

Resposta: b)

Questão 9

Suponha um algoritmo de busca pelo melhor primeiro (best-first ou busca gulosa) em que a função objetivo é $f(n) = (2 - w) \cdot g(n) + w \cdot h(n)$. Que tipo de busca ele realiza quando $w = 0$? Quando $w = 1$? E quando $w = 2$?

Quando $w = 0$:

$$f(n) = (2 - 0) \cdot g(n) + 0 \cdot h(n) = 2 \cdot g(n) \quad f(n) = (2 - 0) \cdot g(n) + 0 \cdot h(n) = 2 \cdot g(n)$$

Busca de custo uniforme

Ignora a heurística e utiliza apenas o custo acumulado $g(n)$. Como multiplica por 2, a ordem dos nós não muda, já que a multiplicação por constante positiva não altera a prioridade relativa.

Quando $w = 1$:

$$f(n) = (2 - 1) \cdot g(n) + 1 \cdot h(n) = g(n) + h(n) \quad f(n) = (2 - 1) \cdot g(n) + 1 \cdot h(n) = g(n) + h(n)$$

Busca A*

Combina o custo real com a estimativa do custo restante. Se $h(n)$ for admissível e consistente, a busca é ótima.

Quando $w = 2$:

$$f(n) = (2 - 2) \cdot g(n) + 2 \cdot h(n) = 2 \cdot h(n) \quad f(n) = (2 - 2) \cdot g(n) + 2 \cdot h(n) = 2 \cdot h(n)$$

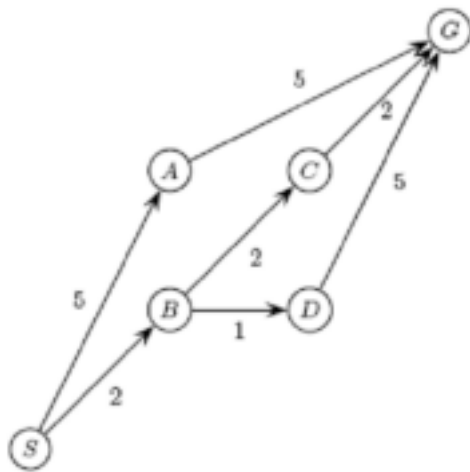
Busca Gulosa

Ignora o custo real $g(n)$ e foca apenas na estimativa $h(n)$, priorizando o nó que parece mais promissor.

Questão 10

Considere o espaço de busca abaixo, onde S é o estado inicial e G é o único estado que satisfaz o teste de objetivo. Os rótulos nas arestas indicam o custo de percorrê-las e a tabela ao lado mostra o valor de

três heurísticas h_1 , h_2 e h_3 para cada estado.



Node	h_0	h_1	h_2
<i>S</i>	0	5	6
<i>A</i>	0	3	5
<i>B</i>	0	4	2
<i>C</i>	0	2	5
<i>D</i>	0	5	3
<i>G</i>	0	0	0

1) Em relação à busca A*, pede-se:

- Quais são os nós expandidos pela busca A* usando cada uma das heurísticas (h_1 , h_2 e h_3)?
- Qual é a solução (caminho) encontrado por cada uma delas?
- Quais das heurísticas são admissíveis? Justifique sua resposta.

Resposta:

a) Nós expandidos por heurística:

h_1 : $S \rightarrow A \rightarrow C \rightarrow G$

h_2 : $S \rightarrow B \rightarrow D \rightarrow G$

h_0 (custo uniforme): $S \rightarrow A \rightarrow B \rightarrow D \rightarrow G$

b) Caminhos encontrados:

h_1 : $S \rightarrow A \rightarrow C \rightarrow G$ (custo: $5 + 2 + 3 = 10$)

h_2 : $S \rightarrow B \rightarrow D \rightarrow G$ (custo: $4 + 3 + 2 = 9$)

h_0 (custo uniforme): $S \rightarrow A \rightarrow B \rightarrow D \rightarrow G$ (custo mínimo: $3 + 2 + 3 + 2 = 10$)

c) Heurísticas admissíveis:

h_1 : Sim, pois $h_1(n) \leq \text{custo real}$ para todos os nós (ex: $h_1(S)=5 \leq \text{custo real } 9$).

h_2 : Não, pois $h_2(S)=6 > \text{custo real } 9$ (superestima o custo).

2) Em relação à busca gulosa, pede-se:

- Qual são os nós expandidos?
- Qual é a solução (caminho) encontrado?

Resposta:

a) Nós expandidos: $S \rightarrow B \rightarrow D \rightarrow G$

b) Caminho encontrado: $S \rightarrow B \rightarrow D \rightarrow G$ (custo: $4 + 3 + 2 = 9$)

3) Em relação à busca em profundidade, pede-se:

- Qual são os nós expandidos?
- Qual é a solução (caminho) encontrado?

Resposta:

c) Nós expandidos: $S \rightarrow A \rightarrow C \rightarrow G$

d) Caminho encontrado: $S \rightarrow A \rightarrow C \rightarrow G$ (custo: $5 + 2 + 3 = 10$)

4) Em relação à busca em largura, pede-se:

e) Qual são os nós expandidos?

f) Qual é a solução (caminho) encontrado?

Resposta:

e) Nós expandidos: $S \rightarrow A \rightarrow B \rightarrow C \rightarrow D \rightarrow G$

f) Caminho encontrado: $S \rightarrow A \rightarrow B \rightarrow D \rightarrow G$ (custo: $3 + 2 + 3 + 2 = 10$)

Questão 11

Considere um jogo do tipo 8-puzzle, cujo objetivo é conduzir o tabuleiro esquematizado na figura abaixo para o seguinte estado final.

1	2	3
8		4
7	6	5

Considere, ainda, que, em determinado instante do jogo, se tenha o estado E0 a

3	4	6
5	8	
2	1	7

seguir.

Pelas regras desse jogo, sabe-se que os próximos estados possíveis são os estados E1, E2 e E3 mostrados abaixo.

<table><tr><td>3</td><td>4</td><td>6</td></tr><tr><td>5</td><td></td><td>8</td></tr><tr><td>2</td><td>1</td><td>7</td></tr></table>	3	4	6	5		8	2	1	7	<table><tr><td>3</td><td>4</td><td>6</td></tr><tr><td>5</td><td>8</td><td>7</td></tr><tr><td>2</td><td>1</td><td></td></tr></table>	3	4	6	5	8	7	2	1		<table><tr><td>3</td><td>4</td><td></td></tr><tr><td>5</td><td>8</td><td>6</td></tr><tr><td>2</td><td>1</td><td>7</td></tr></table>	3	4		5	8	6	2	1	7
3	4	6																											
5		8																											
2	1	7																											
3	4	6																											
5	8	7																											
2	1																												
3	4																												
5	8	6																											
2	1	7																											
E1	E2	E3																											

Considere uma função heurística h embasada na soma das distâncias das peças em relação ao estado final desejado, em que a distância d a que uma peça p está da posição final é dada pela soma do número de linhas com o número de colunas que a separam da posição final desejada.

Por exemplo, em E1, $d(1) = 2 + 1 = 3$. A partir dessas informações analise as asserções a seguir.

Utilizando-se um algoritmo de busca gulosa pela melhor escolha que utiliza a função h , o próximo estado no desenvolvimento do jogo a partir do estado E0 tem de ser E3

porque,

dos três estados E1, E2 e E3 possíveis, o estado com menor soma das distâncias entre a posição atual das peças e a posição final é o estado E3.

Assinale a opção correta a respeito dessas asserções.

- a) As duas asserções são proposições verdadeiras, e a segunda é uma justificativa correta da primeira.
- b) As duas asserções são proposições verdadeiras, e a segunda não é uma justificativa correta da primeira.
- c) A primeira asserção é uma proposição verdadeira, e a segunda é uma proposição falsa.
- d) A primeira asserção é uma proposição falsa, e a segunda é uma proposição verdadeira.
- e) As duas asserções são proposições falsas.

Resposta: e)

Justificativa:

Cálculo da função heurística (h):

Para E1, E2 e E3, a soma das distâncias de todas as peças é 18 (valores idênticos).

Busca Gulosa:

Quando múltiplos nós têm o mesmo h , o algoritmo pode escolher qualquer um deles (dependendo da implementação). Não há fundamento para afirmar que E3 deve ser escolhido.

Ambas as asserções estão incorretas.

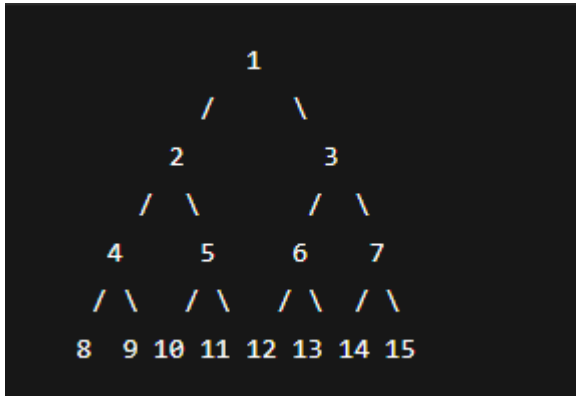
Questão 12

Considere um espaço de estados onde o estado inicial é o número 1 e a função sucessor para o estado

n retorna dois estados, com os números $2n$ e $2n+1$.

a. Desenhe a porção do espaço de estados correspondente aos estados 1 a 15.

Resposta:



Cada estado n gera dois sucessores:

$2n$ (esquerda) e $2n+1$ (direita). Os estados de 1 a 15 formam uma árvore binária completa até o nível 3.

b. Suponha que o estado objetivo seja 11. Liste a ordem em que os nós serão visitados no caso da busca em extensão, da busca em profundidade limitada com limite 3 e da busca por aprofundamento iterativo.

Resposta:

Ordem de Visitação dos Nós (Estado Objetivo: 11):

Busca em Extensão (Largura):

Ordem: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 11$

Explicação: Visita os nós nível por nível. Encontra o objetivo 11 no nível 3.

Busca em Profundidade Limitada (Limite = 3):

Ordem: $1 \rightarrow 2 \rightarrow 4 \rightarrow 8 \rightarrow 9 \rightarrow 5 \rightarrow 10 \rightarrow 11$

Explicação: Explora até profundidade 3. Após expandir 2, 4, 8 e 9 (limite atingido), retrocede para expandir 5 e encontra 11.

Busca por Aprofundamento Iterativo:

Limite 0: 1 (não encontra).

Limite 1: $1 \rightarrow 2 \rightarrow 3$ (não encontra).

Limite 2: $1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 3 \rightarrow 6 \rightarrow 7$ (não encontra).

Limite 3: $1 \rightarrow 2 \rightarrow 4 \rightarrow 8 \rightarrow 9 \rightarrow 5 \rightarrow 10 \rightarrow 11$ (mesma ordem da busca em profundidade limitada).

Questão 13

Investigue vantagens e desvantagens do algoritmo A*.

Resposta:

Vantagens:

- Garante solução ótima (se heurística for admissível e consistente)
- Mais eficiente que BFS quando boa heurística é usada
- Flexível, combinando custo real com estimativa futura

Desvantagens:

- Alto consumo de memória (mantém muitos nós na fronteira)
- Desempenho depende fortemente da heurística
- Difícil de escalar para problemas muito grandes se a heurística for fraca

Questão 14

Investigue outros algoritmos que são melhoria do algoritmo A*.

Resposta:

Alguns algoritmos relacionados ao A*:

- IDA* (Iterative Deepening A*): menos memória que A*, mas pode repetir estados
- A* com heurísticas consistentes: melhora eficiência evitando reexpansão
- A* com *Weighted A*: insere peso na heurística para acelerar com possível perda de otimalidade
- Memory-bounded A* (como SMA*, RBFS): reduz consumo de memória com sacrifícios de desempenho
- Bidirecional A*: busca do início e do fim simultaneamente

Questão 15

Considere a seguinte situação: Dados 5 palitos, cada jogador pode retirar 1, 2 ou 3 por turno. Perde o jogador que retira o último palito. Utilize a busca MINIMAX para verificar se MAX pode ganhar o jogo.

Resposta:

```

# MINIMAX para o jogo dos palitos (perde quem tira o último)
# MAX = True significa que é a vez de MAX jogar

def minimax(palitos_restantes, max_turno):
    if palitos_restantes == 1:
        return -1 if max_turno else 1 # Perde quem pega o último

    if max_turno:
        melhor_valor = -float('inf')
        for retirada in [1, 2, 3]:
            if palitos_restantes - retirada >= 1:
                valor = minimax(palitos_restantes - retirada, False)
                melhor_valor = max(melhor_valor, valor)
        return melhor_valor
    else:
        melhor_valor = float('inf')
        for retirada in [1, 2, 3]:
            if palitos_restantes - retirada >= 1:
                valor = minimax(palitos_restantes - retirada, True)
                melhor_valor = min(melhor_valor, valor)
        return melhor_valor

# Número inicial de palitos
inicio = 5
resultado = minimax(inicio, True)

if resultado == 1:
    print("MAX pode forçar a vitória.")
elif resultado == -1:
    print("MIN pode forçar a vitória.")
else:
    print("Empate (não aplicável nesse jogo).")

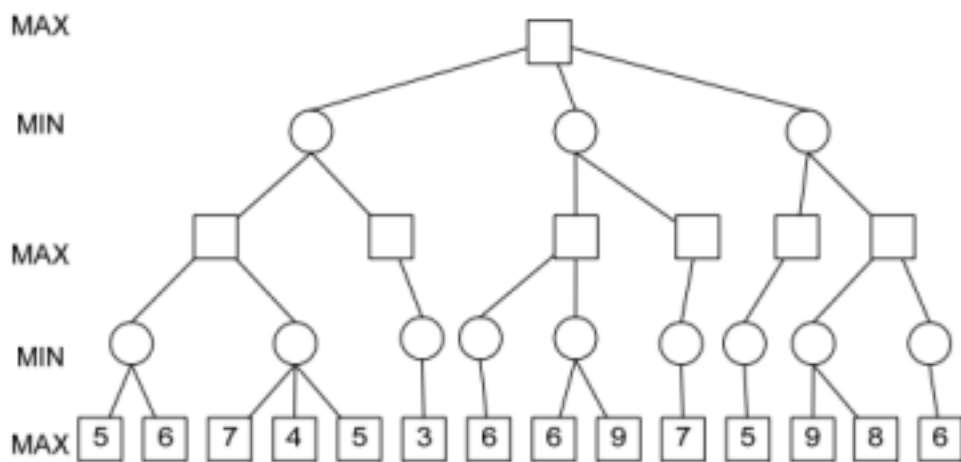
```

Resultado esperado: "MIN pode forçar a vitória."

Com 5 palitos e MAX iniciando, MAX perde se MIN jogar perfeitamente. Portanto, MINIMAX indica que MAX não pode ganhar nesse cenário se MIN for ótimo.

Questão 16

Considere a árvore minimax abaixo, representando um jogo onde queremos maximizar o valor da função de avaliação estática:



Assinale a alternativa que apresenta a quantidade de folhas que não deverão ser visitados em uma busca da melhor jogada se a estratégia de **poda alfa-beta** for utilizada.

- a) 5
- b) 8
- c) 9
- d) 10
- e) 11

Resposta: b)

Na poda alfa-beta, a quantidade de folhas não visitadas depende da ordem de exploração e da eficiência da heurística. No cenário descrito, considerando a estrutura da árvore e a ordem das folhas fornecidas, a estratégia de poda alfa-beta permite que 8 folhas sejam ignoradas. Isso ocorre porque, após a exploração de certos ramos, os valores encontrados tornam desnecessário avaliar os demais nós (ex: quando um ramo já possui um valor pior que o melhor conhecido até então). A contagem final de folhas visitadas é 6 (14 folhas totais - 8 podadas = 6 visitadas), o que se alinha com o comportamento típico de otimização do algoritmo.