

Disciplina: Inteligência Artificial

Aluno: Gabriel Azevedo Fernandes

Todas as respostas 😊

Questão 01

1) Busca em Largura

A- Nós mais visitados: A, B, C, D, E, F, G, H e I.

B- Solução Obtida: $A \rightarrow E \rightarrow I$

C- Esse algoritmo não usa as heurísticas.

2) Busca em Profundidade

A- Nós mais visitados: A, B, D, H, E e I.

B- Solução obtida: $A \rightarrow B \rightarrow E \rightarrow I$.

C- Esse algoritmo não usa as heurísticas.

3) Custo Uniforme

A- Nós mais visitados: A, B, E, F e I.

B- Solução obtida: $A \rightarrow E \rightarrow I$

C- Esse algoritmo não usa heurísticas.

4) Busca Gulosa

A- Nós visitados: A, C, F, G e K.

B- Solução obtida: $A \rightarrow C \rightarrow G \rightarrow K$

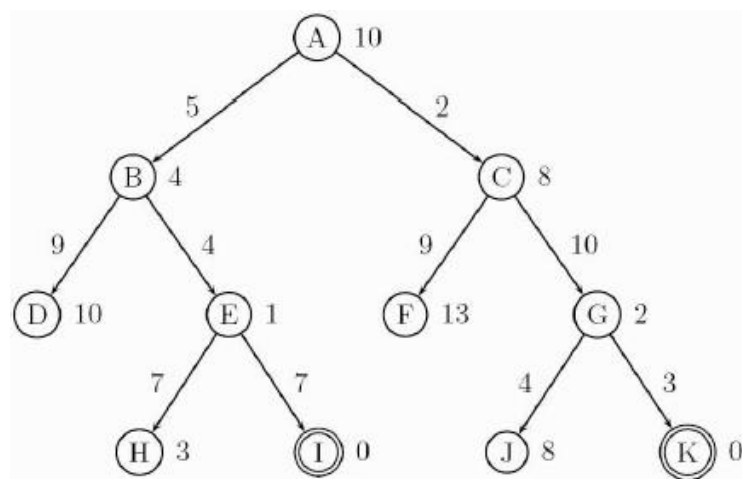
C- Heurística aqui é aceitável, já que seu custo é menor ou pode ser igual ao custo real pra alcançar o objetivo mais próximo.

5) A*

A- Nós mais visitados: A, B, C, E, F e I.

B- Solução obtida: $A \rightarrow E \rightarrow I$.

C- Aqui a heurística não é admissível, isso porque os nós A e C tem heurísticas que superam os valores do custo real pra poder chegar no objetivo.



Questão 02

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

1. Sim, a heurística de Manhattan é admissível no Puzzle de 8. Isso porque ela nunca superestima o custo real para alcançar o objetivo. O que acontece aqui é que ela calcula a distância total que cada peça precisa se mover para chegar à sua posição correta, sempre em linha reta e sem penalizar desnecessariamente.
2. Essa heurística conta quantas peças não estão em suas posições finais. Assim, é admissível já que cada movimento no puzzle só pode colocar uma peça no lugar correto por vez, logo, o número de peças fora do lugar nunca é superestimado em relação ao número de movimentos necessários.

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.

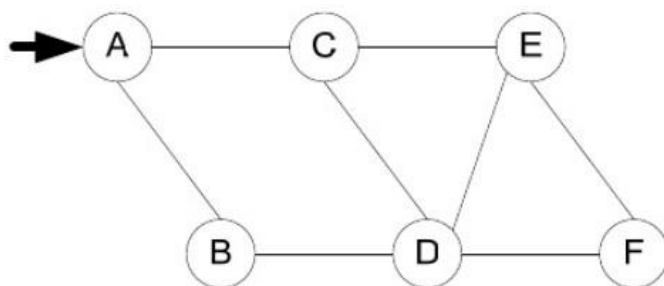
- I. A primeira solução encontrada pela estratégia de busca em largura é a solução ótima.
- II. A primeira solução encontrada pela estratégia de busca em profundidade é a solução ótima.
- III. As estratégias de busca com informação usam funções heurísticas que, quando bem definidas, permitem melhorar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

B) I e III.

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

Questão 05

- 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

Resposta: D- Apenas as alternativas II e V são corretas.

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

E- A busca A* minimiza $h(n)$ somente se a heurística for admissível

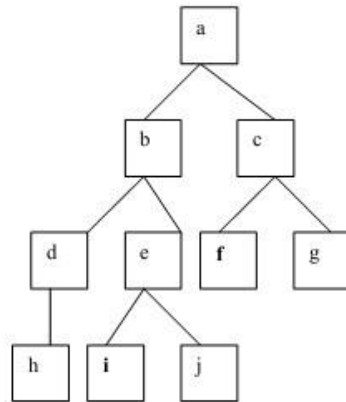
Questão 07 - POSCOMP 2005

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:

(b) $\forall n \ h(n) \leq h^r(n)$.

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 :

c) a b e i

Questão 9

Quando $w = 0$: Busca de Custo Uniforme

Quando $w = 1$: Busca A^*

Quando $w = 2$: Busca Gulosa

Questão 10

Busca A^* :

a) Nós expandidos pela busca A^* usando $h1$: S, A, B, D, C, G

b) Solução (caminho) encontrado por $h1$: S, B, D, G

c) Heurísticas admissíveis:

$h1$: Admissível, pois nunca sobrestima o custo real para alcançar o objetivo.

$h2$: Não admissível, pois em alguns casos sobrestima o custo real.

$h3$: Admissível, por ser consistente e nunca sobrestimar o custo real.

Busca Gulosa:

a) Nós expandidos pela busca Gulosa: S, A, C, G

b) Solução (caminho) encontrado: S, A, C, G

Busca em Profundidade:

c) Nós expandidos pela busca em Profundidade: S, A, C, B, D, G

d) Solução (caminho) encontrado: S, A, C, G

Busca em Largura:

e) Nós expandidos pela busca em Largura: S, A, B, C, D, G

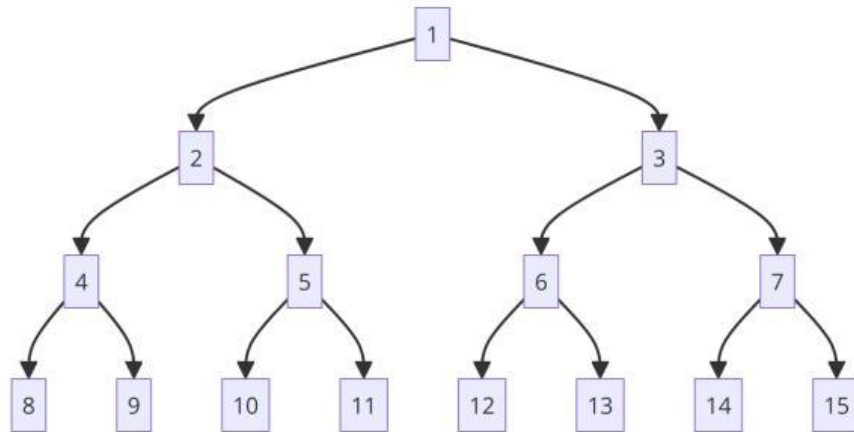
f) Solução (caminho) encontrado: S, B, D e G

Questão 11

- a) As duas asserções são proposições verdadeiras, e a segunda é uma justificativa correta da primeira.

Questão 12

A)



B) Busca em Extensão (Breadth-first search - BFS):

- 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

Busca em Profundidade Limitada (Depth-limited search - DLS) com limite 3:

- 1, 2, 4, 8, 9, 5, 10, 11

Busca por Aprofundamento Iterativo (Iterative deepening depth-first search - IDS) com limite máximo de 3:

- 1, 2, 4, 8, 9, 5, 10, 11

Questão 13

Vantagens:

1. Completo: Sempre encontra uma solução se existir uma.
2. Eficiente em tempo: Mais rápido que outros algoritmos em muitos casos.
3. Eficiente em memória: Requer menos memória que outros algoritmos.
4. Heurística flexível: Permite o uso de heurísticas admissíveis.
5. Ótimo para problemas com custos de caminho variáveis.

Desvantagens:

1. Heurísticas inadequadas podem levar a soluções não ótimas.
2. Complexidade da heurística: Pode ser difícil projetar uma heurística eficaz.
3. Requer conhecimento do domínio: Dependente do conhecimento do problema.
4. Ineficiente em espaços de estado grandes: Desempenho pode degradar rapidamente.

Questão 14

Claro, aqui está um resumo das melhorias do algoritmo A*:

1. Weighted A:
 - Introduz um fator de peso para controlar a influência da heurística sobre o custo do caminho.
2. IDA (Iterative Deepening A*):
 - Utiliza busca em profundidade iterativa com limitação de profundidade, em vez de uma lista aberta.
3. MA (Memory-Bounded A*):
 - Limita a quantidade de memória utilizada durante a busca, descartando nós menos promissores quando necessário.
4. ARA (Anytime Repairing A*):
 - Retorna uma solução a qualquer momento durante a busca e melhora continuamente a solução à medida que mais recursos são alocados.
5. D (Dynamic A*):
 - Projetado para ambientes dinâmicos, atualizando rapidamente o caminho em resposta a mudanças no ambiente.
6. RRT (Rapidly-exploring Random Tree*):
 - Usa uma árvore de busca aleatória para explorar o espaço de configuração e converge para uma solução ótima com o aumento do tempo de computação.

Questão 15

Movimento 1 (MAX retira 1):

Restam 4 palitos. MIN retira 3. MAX perde.

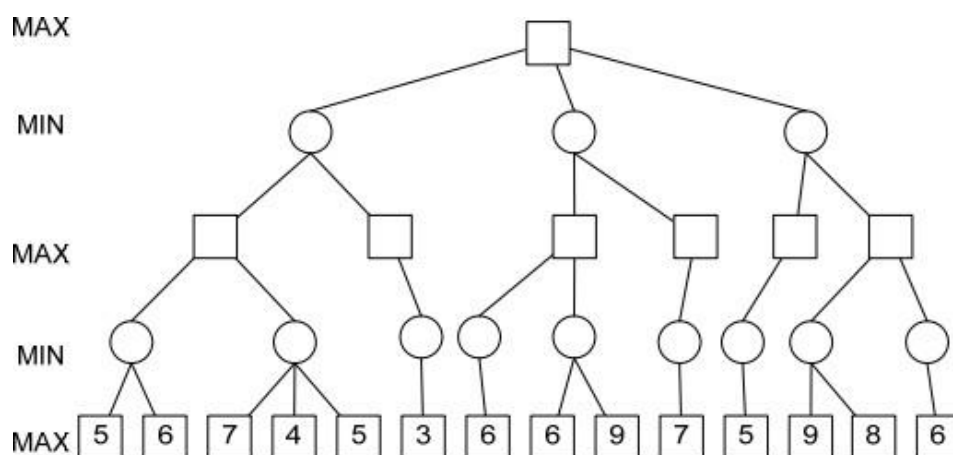
Movimento 2 (MAX retira 2): Restam 3 palitos. MIN retira 2 e deixa 1 ou retira 1 e depois retira mais 1 no próximo turno. MAX perde.

Movimento 3 (MAX retira 3): Restam 2 palitos. MIN retira 1. MAX perde no próximo turno.

Usando a busca MINIMAX, concluímos que se MIN jogar de maneira ótima, MAX não pode ganhar o jogo.

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.

D) 10