

# Lista de IA #8

## Questão 1)

1- Algoritmo de Busca em Largura

- 1) A - B - C - D - E - F - G - H - I
- 2) Ele passa por todos os filhos de um nó, repetindo até passar por todos os vértices.
- 3) Heurística não é empregada

2 - Algoritmo de Busca em Profundidade

- 1)
- 2)
- 3)

**Questão 2)** 1) Sim, pois a heurística de Manhattan calcula a soma de movimentos que devem ser feitos para alcançar o resultado final.

2)

**Questão 3) b)** I e III.

**Questão 4) a)** A B C D E F

**Questão 5) e)** I, IV, V são corretas.

**Questão 6) a)** a busca gulosa minimiza  $h(n)$ .

**Questão 7) b)**  $\forall n \rightarrow h(n) \leq h^*(n)$

**Questão 8) d)** a c f

**Questão 9)**  $w=0 \rightarrow$  Custo Uniforme |  $w=1 \rightarrow A^*$  |  $w=2 \rightarrow$  Busca gulosa

**Questão 10)** 1- a)  $h_0 = \{S, B, D, C, A, G\}$  |  $\{S \rightarrow B \rightarrow C \rightarrow G\}$  | Admissível (trivial)

$h_1 = \{S, B, C, G\}$  |  $\{S \rightarrow B \rightarrow C \rightarrow G\}$  | Admissível (não sobreestima)

$h_2 = \{S, B, D, G\}$  |  $\{S \rightarrow B \rightarrow D \rightarrow G\}$  | Não, pois o custo de  $h_2(C) = 5$  e o custo real é de  $C \rightarrow G = 2$

2-a) Expandidos: S, A, G

b) Caminho :  $S \rightarrow A \rightarrow G$

3-a) Expandidos: S,A,C,G

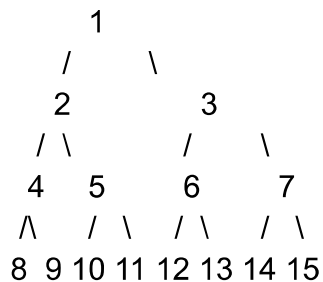
b) Caminho :  $S \rightarrow A \rightarrow C \rightarrow G$

4-a) Expandidos: S,A,B,C,G

b) Caminho :  $S \rightarrow A \rightarrow G$

**Questão 11)** a) As duas asserções são proposições verdadeiras, e a segunda é uma justificativa correta da primeira.

**Questão 12)** a)



b) Largura: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

Profundidade: 1, 2, 4, 8, 9, 5, 10, 11

Aprofundamento Iterativo:

1;

1, 2, 3;

1, 2, 4, 5, 3, 6, 7;

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

**Questão 13)** O algoritmo  $A^*$  (A-estrela) é uma busca heurística que combina os pontos fortes do Dijkstra e da busca gulosa, garantindo completude e ótimo caminho quando um heurístico admissível é utilizado pela Wikipedia. Com um bom heurístico,  $A^*$  torna-se muito eficiente, expandindo substancialmente menos nós em comparação às buscas cegas Medium. Porém, essas vantagens vêm acompanhadas de um alto custo de memória e complexidade de tempo no pior caso, tornando seu uso em grafos muito grandes uma tarefa desafiadora.

**Questão 14)** Existem diversas variantes do  $A^*$  que buscam mitigar suas limitações de memória, adaptar-se a ambientes dinâmicos ou acelerar buscas em grades

regulares. Entre elas destacam-se o IDA\* e o SMA\* para redução de memória, o D\* e suas versões Lite/Anytime para replanejamento em tempo real, heurísticos inflacionados como Weighted A\* e a família Anytime A\* para otimização progressiva, além de algoritmos especializados em grades como Jump Point Search e Theta\*. Cada variante equilibra de forma distinta requisitos de tempo, memória e qualidade de solução.

**Questão 15)** Aplicando a busca Minimax ao jogo de 5 palitos onde cada jogador pode retirar 1, 2 ou 3 por turno e quem retira o último palito perde, obtemos que o estado inicial é uma posição de derrota para o jogador MAX, assumindo jogo ótimo de ambos os lados. A análise detalhada do game tree revela que, para cada escolha de MAX (removendo 1, 2 ou 3 palitos), o jogador MIN possui um movimento subsequente que conduz MAX a uma condição de ser forçado a retirar o último palito e, portanto, perder o jogo.

**Questão 16)** a) 5