

### Questão 1 – Comparação de Estratégias de Busca em Espaço com Heurística

1) Busca em Largura (BFS):

**Visitados:** A, B, C, D, E, F, G, H, I

**Solução:**  $A \rightarrow B \rightarrow E \rightarrow I$

**Heurística não admissível:** Sim

2) Busca em Profundidade (DFS):

**Visitados:** A, B, D, E, H, I

**Solução:**  $A \rightarrow B \rightarrow E \rightarrow I$

**Heurística não admissível:** Sim

3) Custo Uniforme (UCS):

**Visitados:** A, C, B, G, E, K

**Solução:**  $A \rightarrow C \rightarrow G \rightarrow K$

**Heurística não admissível:** Sim

4) Busca Gulosa ( $h(n)$ ):

**Visitados:** A, C, G, K

**Solução:**  $A \rightarrow C \rightarrow G \rightarrow K$

**Heurística não admissível:** Sim

5) Algoritmo A\*:

**Visitados:** A, B, C, E, G, I

**Solução:**  $A \rightarrow B \rightarrow E \rightarrow I$

**Heurística não admissível:** Sim

### Questão 2 – Heurísticas Admissíveis no Puzzle de 8

1) Sim, a heurística de Manhattan é admissível, desde que seja usada em um problema onde o custo real dos movimentos está de acordo com a suposição da heurística. Ela é **admissível** quando **nunca superestima o custo real** do caminho do estado atual até o objetivo ou seja, ela é **otimista**.

2) Outra heurística admissível é o número de peças fora do lugar. Essa heurística simplesmente **conta quantas peças estão fora da posição correta** comparando o estado atual com o estado objetivo. E sim, essa heurística é admissível, e aqui está o porquê:

- Ela nunca superestima o número mínimo de movimentos necessários para resolver o puzzle.
- No pior caso, cada peça fora do lugar precisa de pelo menos um movimento para chegar à posição correta.
- Portanto, ela é otimista, o que a torna admissível.

Questão 3

letra B.

Questão 4

A ordem correta de descoberta dos vértices em busca em largura é:  $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F$ . Logo, a alternativa correta é a (A). “ABCDEF”

Questão 5

Letra E.

Questão 6

Letra A.

Questão 7

Letra B.

Questão 8

Letra C.

Questão 9 –

Quando  $w = 0$ ,  $f(n) = 2g(n)$ , é realizada a busca de custo uniforme, pois apenas o custo do caminho até o momento é considerado.

Quando  $w = 1$ ,  $f(n) = g(n) + h(n)$ , é realizada a busca  $A^*$ .

Quando  $w = 2$ ,  $f(n) = -g(n) + 2h(n)$ , isso dá mais peso à heurística, podendo se comportar de forma semelhante à busca gulosa, mas com a influência negativa do custo do caminho.

Questão 10

1) Algoritmo  $A^*$ : **Nós expandidos:**

- **h1:** S(5), A(8), B(6), D(6), G(6)
- **h2:** S(6), B(4), D(3), G(5)
- **h3:** S(7), B(4), C(7), D(4), G(4) **Solução:** S  $\rightarrow$  B  $\rightarrow$  D  $\rightarrow$  G (para todas as heurísticas)

**Admissibilidade:** h1 e h2 são admissíveis, pois nunca superestimam o custo real do caminho até G. h3 não é admissível porque  $h3(S) = 6$ , mas o custo real é 3.

2) Busca Gulosa ( $h(n)$ ): **expandidos:** Usa menor  $h(n)$ : S( $h=6$ )  $\rightarrow$  B( $h=2$ )  $\rightarrow$  C( $h=5$ )  $\rightarrow$  G( $h=0$ )

**Solução:** S  $\rightarrow$  B  $\rightarrow$  C  $\rightarrow$  G

3) Busca em Profundidade (DFS): **expandidos:** S, A, C, G

**Solução:** S  $\rightarrow$  A  $\rightarrow$  C  $\rightarrow$  G

4) Busca em Largura (BFS): **expandidos:** S, A, B, C, D, G

**Solução:** S  $\rightarrow$  B  $\rightarrow$  D  $\rightarrow$  G

Questão 11

Letra A.

Cálculo das heurísticas:

- E1:  $h(E1) = 3 + 1 + 1 + 1 + 1 + 2 + 0 + 0 = 9$
- E2:  $h(E2) = 3 + 1 + 1 + 0 + 1 + 2 + 0 + 0 = 8$
- E3:  $h(E3) = 3 + 1 + 1 + 0 + 1 + 1 + 0 + 0 = 7$

Questão 12

Busca em largura: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

**Busca em profundidade limitada (limite = 3):** 1, 2, 4, 8, 9, 5, 10, 11

**Aprofundamento iterativo:** Nível 0: 1

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

### Questão 13 – Vantagens e Desvantagens do Algoritmo A\*

#### Vantagens do A\*:

- **Vantagem:** A\* é **completo**, ou seja, **sempre encontra uma solução se ela existir**, desde que o espaço de estados seja finito e os custos dos passos sejam positivos.
- **Vantagem:** A\* é muito mais eficiente do que buscas como a **busca em largura** ou **busca de custo uniforme** porque **usa a heurística para guiar a exploração**, focando nas regiões promissoras do espaço de busca.

#### Desvantagens do A\*:

1. A\* precisa manter na memória **todos os nós gerados** (na fronteira e visitados) para garantir completude e otimalidade.
2. Mesmo com heurísticas boas, se o espaço de busca for grande ou a heurística for pouco informativa, o número de nós expandidos ainda pode ser enorme.

### Questão 14

#### IDA\* (Iterative Deepening A\*)

- Usa aprofundamento iterativo para limitar o uso de memória.

#### ARA\* (Anytime Repairing A\*)

- **Como funciona:** É um algoritmo "**anytime**", ou seja, fornece uma solução rápida subótima inicialmente e a **melhora com o tempo** se mais tempo for dado.

#### SMA\* (Simplified Memory-Bounded A\*)

- Usa uma quantidade limitada de memória e descarta os piores nós quando a memória está cheia.

#### A\* com Heurísticas Ponderadas

- Pondera a função heurística para dar mais ou menos importância à estimativa de custo.

### Questão 15

### Questão 16

Durante a execução do algoritmo Minimax com poda alfa-beta, nem todos os nós folha precisam ser visitados. Isso acontece porque, à medida que percorremos a árvore da esquerda para a direita, o algoritmo consegue eliminar ramos que não influenciam na decisão final, poupando esforço computacional. Na árvore apresentada, esse processo permite evitar a visita a 10 folhas, pois os valores já obtidos foram suficientes para descartar outros caminhos que não ofereceriam melhores resultados. Por isso, a alternativa correta é a letra (D).