Lista IA

Questão 1:

1. Busca em Largura (BFS)

- Explora os nós por nível.
- Caminho mais curto em número de passos, não necessariamente o de menor custo.

Ordem de visitação:

S, A, B, C, D, E, F, G, H, I, J, Z

Caminho encontrado:

S → C → G → J → Z

---

2. Busca em Profundidade (DFS)

- Explora o caminho mais profundo primeiro.
- Depende da ordem dos filhos (seguiremos da esquerda para a direita conforme a imagem).

Ordem de visitação:

S, A, D, E, H, I, B, F, C, G, J, Z

Caminho encontrado:

S → C → G → J → Z

(mesmo caminho do BFS, mas chega até lá depois de explorar mais profundamente por outros ramos)

---

3. Busca de Custo Uniforme

- Expande o nó com menor custo acumulado (g(n)).
- Garante o menor custo total da raiz ao objetivo.

Ordem de visitação:

S, A, B, C, D, E, F, G, H, I, J, Z

(a ordem pode variar levemente dependendo dos custos, mas o importante é seguir os menores custos acumulados)

Caminho encontrado (menor custo acumulado):

S → A → E → I → Z

---

4. Busca Gulosa

- Usa apenas a heurística (h(n)) para decidir o próximo nó.
- Não garante o melhor caminho, mas tenta se aproximar rápido do objetivo.

Ordem de visitação:

S, C, G, J, Z

(supondo que os valores heurísticos façam G parecer próximo de Z)

Caminho encontrado:

S → C → G → J → Z

---

5. A

* (A Estrela)

- Usa: f(n) = g(n) + h(n)
  - g(n): custo acumulado até o nó.
  - h(n): heurística (estimativa até Z).
- Garante o melhor caminho, se a heurística for admissível.

Ordem de visitação:

S, A, B, E, I, Z

Caminho encontrado (ótimo custo + heurística):

S → A → E → I → Z

Questão 2:

1. A heurística de Manhattan é admissível? Justifique.

Sim, a heurística de Manhattan é admissível.

Justificativa:

- A heurística de Manhattan calcula a soma das distâncias horizontais e verticais de cada peça até sua posição correta.
- Ela nunca superestima o número real de movimentos necessários, pois:
  - Cada movimento no puzzle só pode mover uma peça por vez e em uma direção (cima, baixo, esquerda ou direita).

- Como não conta movimentos “extras”, a estimativa é sempre menor ou igual ao número real de passos necessários.

Logo, ela é admissível (nunca superestima) e também consistente, o que a torna adequada para o algoritmo A*.

---

2. Proponha uma outra heurística para este problema. Ela é admissível? Justifique.

Outra heurística:

Número de peças fora do lugar (também chamada de “peças deslocadas”).

Como funciona:

- Conta quantas peças não estão na posição correta.
- Exemplo: Se 3 peças estão fora do lugar, a heurística retorna 3.

É admissível?

Sim, é admissível.

Justificativa:

- O número de peças fora do lugar não superestima a quantidade mínima de movimentos.
- Na pior das hipóteses, será necessário pelo menos um movimento por peça fora do lugar, então a estimativa é válida.
- Porém, é menos informativa que a de Manhattan (menos precisa), o que pode levar o A* a expandir mais nós.

Questão 3:

b) I e III.

Questão 4:

a) A B C D E F

Questão 5:

e) I, IV e V

Questão 6:

a) a busca gulosa minimiza $h(n)$

Questão 7:

b) Vn $h(n) \leq h^{\circ}(n)$.

Questão 8:

e) a b d e f

Questão 9:

1. Quando $w = 0$:

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

   → A função considera apenas o custo real, multiplicado por 2 (constante).

   Equivale à busca de custo uniforme (pois só considera $g(n)$).

2. Quando $w = 1$:

   $f(n) = g(n) + h(n)$

→ Isso é exatamente a fórmula da busca A*.

(ótima se a heurística for admissível)

3. Quando w = 2:

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

   → Só considera a heurística.

   Equivale à busca gulosa (best-first search), já que só minimiza h(n).

Questão 10:

1a)

- h1: S, A, C, G
- h2: S, A, B, C, G

1b)

- Ambos: S → A → C → G

1c)

- h1 e h2 são admissíveis

2a)

- Nós expandidos: S, B, A, C, G

2b)

- Caminho: S → A → C → G

3c)

- Nós expandidos: S, A, C, G

3d)

- Caminho: S → A → C → G

4e)

- Nós expandidos: S, A, B, C, G

4f)

- Caminho: S → A → C → G

Questão 11:

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

Questão 12:

a)   1

          /   \

```
       2       3
      / \     /  \
     4   5   6    7
    / \    / \   / \    / \
   8  9 10 11 12 13 14 15
```

b) Ordem de visita aos nós:

- Busca em largura (BFS):

Visita nível por nível:

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

- Busca em profundidade limitada (limite = 3):

Segue o caminho mais à esquerda até o limite:

1, 2, 4, 8, 9, 5, 10, 11 (encontrou o objetivo no nível 3)

- Aprofundamento iterativo (DFS com profundidade crescente):

- Profundidade 0: 1
- Profundidade 1: 1, 2, 3
- Profundidade 2: 1, 2, 4, 5, 3, 6, 7
- Profundidade 3: 1, 2, 4, 8, 9, 5, 10, 11 (encontrou o 11)

Questão 13:

Vantagens:

- Garante encontrar a solução ótima se a heurística for admissível.
- Muito eficiente em termos de nós expandidos com boas heurísticas.
- Flexível ao usar f(n) = g(n) + h(n).

Desvantagens:

- Alto uso de memória (mantém todos os nós abertos em memória).
- Pode ser lento se a heurística não for informativa.
- Custo computacional pode ser alto em grafos grandes.

Questão 14:

Algoritmos derivados de A:*

- IDA* (Iterative Deepening A*): usa menos memória, combina A* com aprofundamento iterativo.
- SMA* (Simplified Memory-Bounded A*): otimiza uso de memória, removendo nós com menor prioridade quando necessário.
- RBFS (Recursive Best-First Search): usa recursão para manter o uso de memória baixo.
- WA* (Weighted A*): sacrifica otimalidade em troca de velocidade com f(n) = g(n) + w*h(n).

Questão 15:

Estado inicial: 5 palitos. Jogadores alternam retirando 1, 2 ou 3.

Estados possíveis:

- MAX joga: 5 → (4, 3, 2)
- MIN joga:

    - 4 → (3, 2, 1)
    - 3 → (2, 1, 0)
    - 2 → (1, 0)
- MAX joga:
    - 1 → (0) — PERDE quem joga aqui
    - 0 → terminal

Aplicando Minimax:

- Se MAX começa com 5:
    - Se ele tira 2 palitos, fica 3 para MIN.
    - MIN terá opções:
        - Tirar 1 → 2 palitos para MAX
        - Tirar 2 → 1 palito para MAX
        - Tirar 3 → 0 palitos (perde)
    - MAX pode vencer se MIN deixar 1 palito.

Conclusão:

Sim, MAX pode vencer, com a jogada inicial correta.

Questão 16:

b) 8