**Lista 4 - Inteligência Artificial**

**André Mendes Rodrigues – 780371**

**QUESTÃO 1 -**

1) Algoritmo de Busca em Largura:

Nós visitados na ordem: A, B, C, D, E, F, G, H, I

Solução obtida: A → B → D → I

Heurística admissível? Sim, porque nunca superestima o custo real para alcançar o objetivo.

2) Algoritmo de Busca em Profundidade:

Nós visitados na ordem: A, B, D, I

Solução obtida: A → B → D → I

Heurística admissível? Não é aplicável, pois este algoritmo não usa heurísticas.

3) Custo Uniforme:

Nós visitados na ordem: A, B, C, D, E, F, G, I

Solução obtida: A → B → D → I

Heurística admissível? Sim, a busca é sempre ótima, pois expande os nós com menor custo.

4) Algoritmo de Busca Gulosa:

Nós visitados na ordem: A, B, C, D, I

Solução obtida: A → B → D → I

Heurística admissível? Sim, pois ela guia a busca de forma eficiente ao objetivo.

5) Algoritmo A\*:

Nós visitados na ordem: A, B, D, I

Solução obtida: A → B → D → I

Heurística admissível? Sim, pois nunca superestima o custo.

**QUESTÃO 2 -**

1) Sim, pois ela nunca superestima o número de movimentos restantes no Puzzle de 8

2) A contagem de peças fora do lugar é admissível, pois também nunca superestima o custo real para alcançar o objetivo.

**QUESTÃO 3 - (B) I e III**

A alternativa correta é “b) I e III”, pois tanto a busca em largura quanto o algoritmo A\* garantem a solução ótima, desde que a heurística de A\* seja admissível.

**QUESTÃO 4 - (B) A B D C E F**

Na busca em largura, visitamos todos os vizinhos de um nó antes de explorar os nós em níveis mais profundos. Começando por A, seus vizinhos diretos são B e C, com B sendo descoberto primeiro. Em seguida, exploramos B, que não tem novos vizinhos, e passamos para C, cujos vizinhos D e E são visitados. Por fim, exploramos D, que tem o vizinho F.

**QUESTÃO 5 - (E) apenas as afirmativas I, IV e V são corretas**

II: A busca em profundidade não necessariamente explora menos nós que a busca em largura.

III: A busca heurística não garante encontrar a solução ótima, a menos que a função heurística seja admissível.

**QUESTÃO 6 - (A) a busca gulosa minimiza h(n).**

A busca gulosa, por definição, sempre escolhe o nó com o menor valor de h(n), buscando minimizar essa estimativa.

**QUESTÃO 7 - (B) ∀n h(n) ≤ h** ® **(n).**

Uma heurística é admissível quando a estimativa do custo nunca é maior que o custo real para todos os nós.

**QUESTÃO 8 - (C)** **a b e i**

A alternativa C representa a ordem exata em que os nós da árvore são visitados pela busca em largura-primeira, começando do nó inicial "a" e encontrando o objetivo "i" no menor número de passos possível.

**QUESTÃO 9 -**

A função de avaliação f(n) = (2 - w) \* g(n) + w \* h(n) gera diferentes algoritmos dependendo do valor de **w**:

**w = 0:** Corresponde à busca de custo uniforme, pois considera apenas o custo g(n).

**w = 1:** É o algoritmo A\*, equilibrando o custo g(n) e a heurística h(n).

**w = 2:** Corresponde à busca gulosa, que prioriza apenas a heurística h(n).

**QUESTÃO 10 -**

1) A\* com heurísticas h1, h2 e h3:

(a) Nós expandidos: Dependendo da heurística, a ordem de expansão muda. Precisa ser calculado para cada uma.

(b) Solução: A solução será o caminho que minimiza o custo total g(n) + h(n).

(c) Admissibilidade: A heurística é admissível se nunca superestimar o custo para alcançar o objetivo.

2) Busca gulosa:

(a) Expande os nós de acordo com a menor heurística h(n).

(b) A solução é o caminho com menor valor de h(n), mas pode não ser o mais curto.

3) Busca em profundidade:

(c) Expande os nós até alcançar uma folha.

(d) A solução pode não ser a ótima, pois depende de quando o objetivo é encontrado.

4) Busca em largura:

(e) Expande nós na ordem em que foram descobertos.

(f) A solução é o caminho mais curto em termos de número de nós.

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

A admissibilidade da heurística é a chave para a solução ótima no A\*.

**QUESTÃO 12 -**

**a) 1**

**/ \**

**2 3**

**/ \ / \**

**4 5 6 7**

**/ \ / \ / \ / \**

**8 9 10 11 12 13 14 15**

**b)**

1. Busca em Largura:

A busca em largura explora o nível completo de cada nó antes de passar para os nós de nível mais profundo. A ordem de visitação dos nós é:

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

Ou seja, a ordem completa será: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11.2. Busca em Profundidade Limitada com limite 3

Na busca em profundidade limitada, os nós são explorados até o nível 3 da árvore, com uma profundidade máxima de 3. Segue-se uma exploração profunda de cada ramo até o limite. A ordem de visitação dos nós é:

1 → 2 → 4 → 8 → 9 (fim do ramo)

Volta para o próximo ramo:

1 → 2 → 5 → 10 → 11

Ou seja, a ordem completa será: 1, 2, 4, 8, 9, 5, 10, 11

3. Busca por Aprofundamento Iterativo

Na busca por aprofundamento iterativo, começamos explorando em profundidade com limite 1, depois aumentamos o limite para 2, e assim por diante, até encontrar o estado objetivo. A cada nova profundidade, o processo é reiniciado. A ordem de visitação será:

Limite 1: 1

Limite 2: 1 → 2 → 3

Limite 3: 1 → 2 → 4 → 5 → 3 → 6 → 7

Limite 4: 1 → 2 → 4 → 8 → 9 → 5 → 10 → 11

Ou seja, a ordem completa será: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11.

**QUESTÃO 13 -**

Vantagens do A\*:

* Garante a solução ótima se a heurística for admissível.
* Mais eficiente que a busca cega.

Desvantagens do A\*:

* Pode consumir muita memória, pois mantém todos os nós expandidos.

**QUESTÃO 14 -**

Algoritmos que aprimoram o A\* incluem o IDA\* e o D\*, que são mais eficientes em termos de memória, utilizando menos recursos enquanto mantêm a admissibilidade e a qualidade da solução.

**QUESTÃO 15 -**

1. Definições:
   1. MAX: jogador que tenta maximizar suas chances de ganhar.
   2. MIN: jogador que tenta minimizar as chances de MAX.
2. Estado Inicial: 5 palitos.
3. Jogadas Possíveis:
   1. Se MAX retirar 1 palito, ficam 4 palitos.
   2. Se MAX retirar 2 palitos, ficam 3 palitos.
   3. Se MAX retirar 3 palitos, ficam 2 palitos.
4. Árvore de Decisão:
   1. Se MAX retirar 1 (4 palitos restantes):
      1. MIN pode retirar:
         1. 1 (3 palitos restantes)
         2. 2 (2 palitos restantes)
         3. 3 (1 palito restante)
      2. Analisando cada uma das jogadas de MIN:
         1. 3 palitos: MAX pode ganhar retirando 1 ou 2 (MAX ganha)
         2. 2 palitos: MAX pode ganhar retirando 2 (MAX ganha)
         3. 1 palito: MAX perde (porque MIN vai retirar o último palito).

Então, se MAX retirar 1 palito, ele tem uma jogada que leva à vitória.

* 1. Se MAX retirar 2 (3 palitos restantes):
     1. MIN pode retirar:
        1. 1 (2 palitos restantes)
        2. 2 (1 palito restante)
        3. 3 (0 palitos restantes)
     2. Analisando cada jogada de MIN:
        1. 2 palitos: MAX pode retirar 2 e perder (porque MIN vai retirar o último palito).
        2. 1 palito: MAX pode retirar 1 e ganhar (MAX ganha).
        3. 0 palitos: já acabou.

Aqui, MAX tem uma jogada que leva à vitória.

* 1. Se MAX retirar 3 (2 palitos restantes):
     1. MIN pode retirar:
        1. 1 (1 palito restante)
        2. 2 (0 palitos restantes)
     2. Analisando:
        1. 1 palito: MAX perde (porque MIN vai retirar o último palito).
        2. 0 palitos: já acabou.

Aqui, MAX perde.

1. Resultados:
   1. A partir de 5 palitos, MAX pode garantir a vitória retirando 1 ou 2 palitos, pois em ambas as situações ele pode forçar uma situação em que MIN acaba perdendo.

Usando a estratégia MINIMAX, podemos concluir que MAX pode ganhar o jogo se ele jogar corretamente, retirando 1 ou 2 palitos em seu primeiro turno.

**QUESTÃO 16 - (D) 10**