

**Disciplina: Inteligência Artificial**

**Professora: Cristiane Neri Nobre**

**Data de entrega: 04/05**

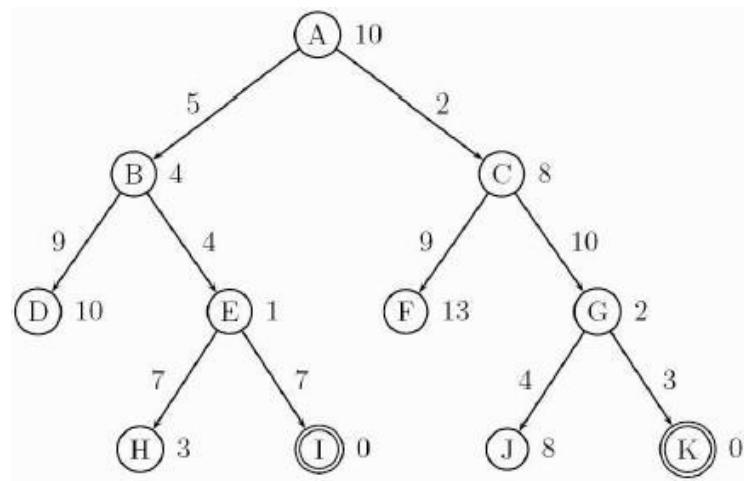
**Valor: 1,5 pontos**

### Questão 01

Considere o espaço de busca a seguir. Cada nó é rotulado por uma letra. Cada nó objetivo é representado por um círculo duplo. Existe uma heurística estimada para cada dado nó (indicada por um valor ao lado do nó). Arcos representam os operadores e seus custos associados. Para cada um dos algoritmos a seguir, pede-se:

- 1) Os **nós visitados** na ordem em que eles são examinados, começando pelo nó A
- 2) Forneça também a **solução obtida** por cada método
- 3) Pergunta-se: a **heurística** é admissível? Justifique.

No caso de escolhas equivalentes entre diferentes nodos, prefira o nodo mais próximo da raiz, seguido pelo nodo mais à esquerda na árvore. O algoritmo pára a busca quando encontra o I ou o K. Ou seja, não é necessário encontrar os dois objetivos.



- 1) Algoritmo de Busca em Largura
  1. A, B, C, D, E, F, G, H, I
  2. A, B, E, I: 16
  3. Não utiliza heurística
- 2) Algoritmo de Busca em Profundidade
  1. A, B, D, E, H, I
  2. A, B, E, I: 16
  3. Não utiliza heurística
- 3) Custo Uniforme
  1. A, C, B, E, F, G, D, K, H, I, J
  2. A, B, E, I: 16 e A, C, G, K: 15
  3. Não utiliza heurística
- 4) Algoritmo de Busca Gulosa
  1. A, B, E, I

2. A, B, E, I: 16
  3. É admissível pois todas as heurísticas calculadas para cada nó são menores que o custo real
- 5) Algoritmo A\*
1. A, B, E, I
  2. A, B, E, I: 16
  3. É admissível pois todas as heurísticas calculadas para cada nó são menores que o custo real

## Questão 02

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

1. A heurística de Manhattan é admissível? Justifique.

A heurística de Manhattan que faz o cálculo das distâncias  $|x_1 - x_2| + |y_1 - y_2|$  retorna o número mínimo de movimentos para acertar cada posição sem considerar colisões com outras peças. Ou seja, é impossível que a heurística atribuída a cada nó seja maior do que o custo real. Portanto é admissível

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

Uma outra heurística poderia ser a Distância euclidiana de uma peça para seu lugar correto. Ela é admissível pois seu valor é ainda menor que a distância de manhattan isso pode ser comprovado por geometria dos triângulos a distância de manhattan é como se fosse os dois lados menores do triângulo e a distância euclidiana é o lado maior do triângulo retângulo, que por definição  $A < B$

### 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.

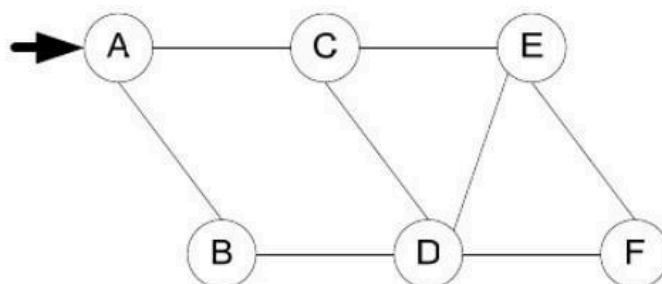
- I. A primeira solução encontrada pela estratégia de busca em largura é a solução ótima.
- II. A primeira solução encontrada pela estratégia de busca em profundidade é a solução ótima.
- III. As estratégias de busca com informação usam funções heurísticas que, quando bem definidas, permitem melhorar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

- a) I e II.
- b) I e III.**
- c) I e IV.
- d) II e IV.
- e) III e IV.

### 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
- B) A B D C E F
- C) A C D B F E
- D) A B C E D F
- E) A B D F E C

resposta: letra a.

### Questão 05

Analisar as seguintes afirmativas:

- 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

- (a) apenas a afirmativa V é correta.
- (b) todas as afirmativas são corretas.
- (c) todas as afirmativas são falsas.
- (d) apenas as afirmativas II e V são corretas.
- (e) apenas as afirmativas I, IV e V são corretas.

resposta: letra e.

#### 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

- (a) a busca *gulosa* minimiza  $h(n)$ .
- (b) a busca  $A^*$  minimiza  $h(n)$ .
- (c) a busca de custo uniforme minimiza  $h(n)$ .
- (d) a busca *gulosa* minimiza  $h(n)$  somente se a heurística for admissível.
- (e) a busca  $A^*$  minimiza  $h(n)$  somente se a heurística for admissível.

resposta: letra a

#### Questão 07 - POSCOMP 2005

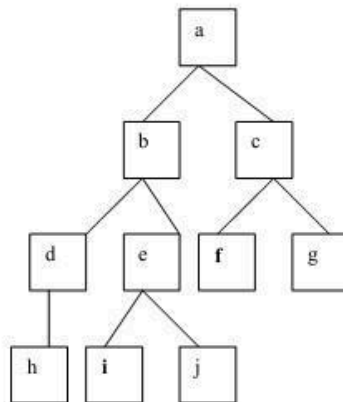
Considere  $h(x)$  como uma função heurística que define a distância de  $x$  até a meta; considere ainda  $h^r(x)$  como a distância real de  $x$  até a meta.  $h(x)$  é dita admissível se e somente se:

- (a)  $\exists n \ h(n) \leq h^r(n)$ .
- (b)  $\forall n \ h(n) \leq h^r(n)$ .
- (c)  $\forall n \ h(n) > h^r(n)$ .
- (d)  $\exists n \ h(n) > h^r(n)$ .
- (e)  $\exists n \ h(n) < h^r(n)$ .

resposta letra b

### 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 seqüência de estados como primeira alternativa a um caminho-solução para o problema p:

- a) a b d h e i
- b) a b c d e f
- c) a b e i
- d) a c f
- e) a b d e f

resposta letra b

### Questão 9

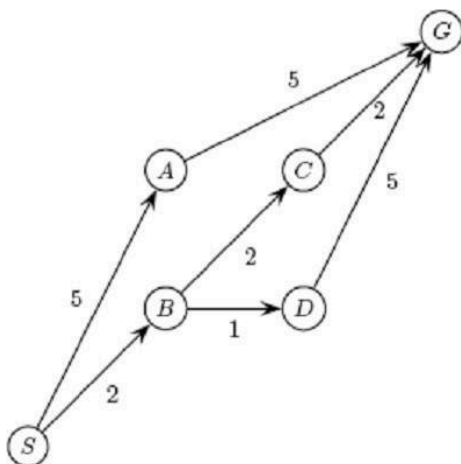
Suponha um algoritmo de busca pelo melhor primeiro (best-first ou busca gulosa) em que a função objetivo é  $f(n) = (2 - w).g(n) + w.h(n)$ . Que tipo de busca ele realiza quando  $w = 0$ ? Quando  $w = 1$ ? E quando  $w = 2$ ?

**Quando  $w = 0$  ele faz a busca de custo uniforme pois considera só o  $g(n)$  já que zera o  $h(n)$ . Quando  $w$  é igual a 1 então,  $f(n) = h(n) + g(n)$ , que é o algoritmo do A\*. Quando  $w = 2$  ai ele faz a busca gulosa normal, pois prioriza somente a heurística já que zera o  $g(n)$**

### Questão 10

Considere o espaço de busca abaixo, onde S é o estado inicial e G é o único estado que satisfaz o teste de objetivo. Os rótulos nas arestas indicam o custo de percorrê-las e a tabela ao lado mostra o valor de

três heurísticas  $h_1$ ,  $h_2$  e  $h_3$  para cada estado.



Node	$h_0$	$h_1$	$h_2$
S	0	5	6
A	0	3	5
B	0	4	2
C	0	2	5
D	0	5	3
G	0	0	0

- 1) Em relação à busca A\*, pede-se:
  - a) Quais são os nós expandidos pela busca A\* usando cada uma das heurísticas (h1, h2 e h3)?  
**h0: S, A, G**  
**h1: S, B, C, G**  
**h2: S, B, D, G**
  - b) Qual é a solução (caminho) encontrado por cada uma delas?  
**h0: S, A, G**  
**h1: S, B, C, G**  
**h2: S, B, D, G**
  - c) Quais das heurísticas são admissíveis? Justifique sua resposta.  
**h0 é admissível pois pelo fato de todos serem zero nunca vai ser maior que a distância real.**  
**h1 também é admissível pois não há um h1(n) maior que o custo real.**  
**h2 não é admissível pois  $h2(C) = 5$  e o custo real é 2.**
  
- 2) Em relação à busca gulosa, pede-se:
  - a) Qual são os nós expandidos?  
**S, A, G**
  - b) Qual é a solução (caminho) encontrado?  
**S, A, G**
  
- 3) Em relação à busca em profundidade, pede-se:
  - c) Qual são os nós expandidos?  
**S, A, G**
  - d) Qual é a solução (caminho) encontrado?  
**S, A, G**
  
- 4) Em relação à busca em largura, pede-se:
  - e) Qual são os nós expandidos?  
**S, A, B, G**
  - f) Qual é a solução (caminho) encontrado?  
**S, A, G**

### Questão 11

Considere um jogo do tipo 8-puzzle, cujo objetivo é conduzir o tabuleiro esquematizado na figura abaixo para o seguinte estado final.

1	2	3
8		4
7	6	5

Considere, ainda, que, em determinado instante do jogo, se tenha o estado E0 a seguir.

3	4	6
5	8	
2	1	7

Pelas regras desse jogo, sabe-se que os próximos estados possíveis são os estados E1, E2 e E3 mostrados abaixo.

<table> <tr><td>3</td><td>4</td><td>6</td></tr> <tr><td>5</td><td></td><td>8</td></tr> <tr><td>2</td><td>1</td><td>7</td></tr> </table>	3	4	6	5		8	2	1	7	<table> <tr><td>3</td><td>4</td><td>6</td></tr> <tr><td>5</td><td>8</td><td>7</td></tr> <tr><td>2</td><td>1</td><td></td></tr> </table>	3	4	6	5	8	7	2	1		<table> <tr><td>3</td><td>4</td><td></td></tr> <tr><td>5</td><td>8</td><td>6</td></tr> <tr><td>2</td><td>1</td><td>7</td></tr> </table>	3	4		5	8	6	2	1	7
3	4	6																											
5		8																											
2	1	7																											
3	4	6																											
5	8	7																											
2	1																												
3	4																												
5	8	6																											
2	1	7																											
E1	E2	E3																											

Considere uma função heurística  $h$  embasada na soma das distâncias das peças em relação ao estado final desejado, em que a distância  $d$  a que uma peça  $p$  está da posição final é dada pela soma do número de linhas com o número de colunas que a separam da posição final desejada.

Por exemplo, em E1,  $d(1) = 2 + 1 = 3$ . A partir dessas informações analise as asserções a seguir.

Utilizando-se um algoritmo de busca gulosa pela melhor escolha que utiliza a função  $h$ , o próximo estado no desenvolvimento do jogo a partir do estado E0 tem de ser E3

porque,

dos três estados E1, E2 e E3 possíveis, o estado com menor soma das distâncias entre a posição atual das peças e a posição final é o estado E3.

Assinale a opção correta a respeito dessas asserções.

- As duas asserções são proposições verdadeiras, e a segunda é uma justificativa correta da primeira.
- As duas asserções são proposições verdadeiras, e a segunda não é uma justificativa correta da primeira.
- A primeira asserção é uma proposição verdadeira, e a segunda é uma proposição falsa.
- A primeira asserção é uma proposição falsa, e a segunda é uma proposição verdadeira.
- As duas asserções são proposições falsas.

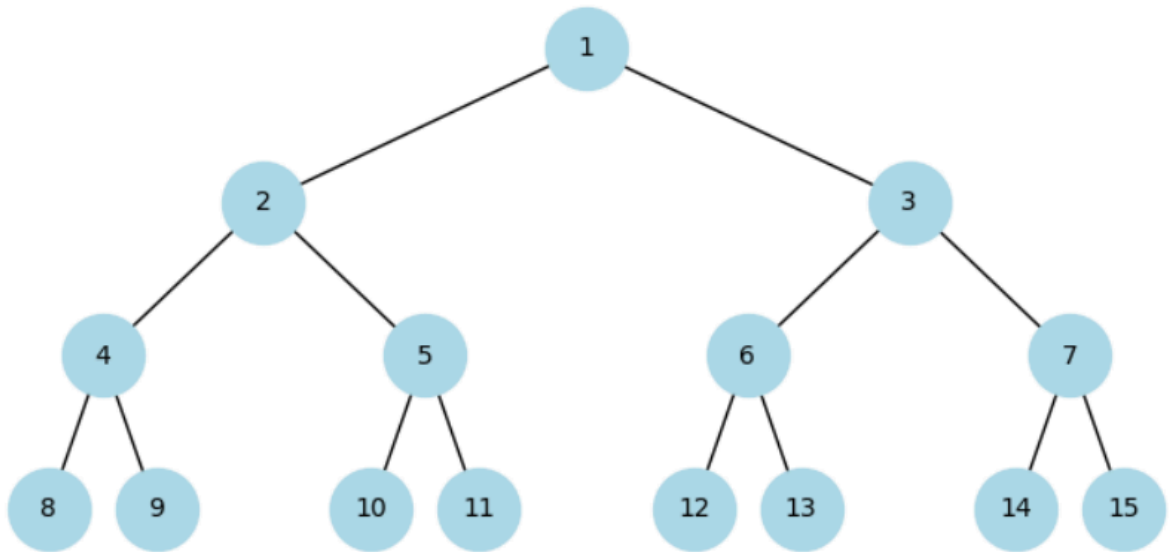
### Questão 12

Considere um espaço de estados onde o estado inicial é o número 1 e a função sucessor para o estado  $n$  retorna dois estados, com os números  $2n$  e  $2n+1$ .

- Desenhe a porção do espaço de estados correspondente aos estados 1 a 15.



Árvore dos estados de 1 a 15



b. Suponha que o estado objetivo seja 11. Liste a ordem em que os nós serão visitados no caso da busca em extensão, da busca em profundidade limitada com limite 3 e da busca por aprofundamento iterativo.

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

**Busca em profundidade limitada igual a 3: 1, 2, 4, 8, 9, 5, 10, 11**

**Busca por aprofundamento iterativo:**

**iteração 0: 1**

**iteração 1: 1, 2, 3**

**iteração 2: 1, 2, 4, 5, 3, 6, 7,**


**iteração 3: 1, 2, 4, 8, 9, 5, 10, 11**

### Questão 13

Investigue vantagens e desvantagens do algoritmo A\*.

#### Vantagem


#### Explicação

 **Solução ótima garantida**

Se a heurística for **admissível** (não superestima), o A\* sempre encontra o caminho mais barato

 **Completo**

Sempre encontra a solução se ela existir (em grafos com custo positivo)

 **Mais eficiente que BFS/DFS**

Usa a heurística  $h(n)$  para **guiar a busca**, evitando expandir caminhos ruins

 **Balanceado**

Leva em conta tanto o **custo já gasto** ( $g(n)$ ) quanto a **estimativa até o fim** ( $h(n)$ )

 **Flexível**

Funciona com qualquer heurística, e pode ser ajustado para diferentes estratégias



**Melhor que a gulosa em precisão**

Mesmo quando mais lento, **não se perde em estimativas ruins** como a busca gulosa

### **Desvantagem**

### **Explicação**



**Uso intensivo de memória**

Guarda todos os nós gerados na fila de prioridade (open list), o que pode ser muito grande



**Desempenho depende da heurística**

Se a heurística for ruim (subestimando demais), ele se comporta como Dijkstra



**Pode ser lento em espaços grandes**

Em espaços de busca enormes, pode expandir muitos nós mesmo sendo guiado



**Implementação mais complexa**

Precisa de estrutura de dados eficiente (como heap), mapa de custos, e controle da fronteira



**Pode reprocessar nós**

Se a heurística for **inconsistente**, pode expandir o mesmo nó mais de uma vez

### Questão 14

Investigue outros algoritmos que são melhoria do algoritmo A\*

1. **A\* com aprofundamento iterativo:** Usa o mesmo algoritmo do a\* porém a cada iteração é estabelecido um limite máximo de  $f(n)$ , os vértices que não respeitarem esse limite serão ignorados. A cada iteração o limite vai aumentando. A vantagem é que evita guardar todos os nós na memória.
2. **A\* ponderado:** usa a fórmula  $f(n) = g(n) + w * h(n)$  aumentando a influência da heurística conforme o peso. É útil quando se aceita uma solução não ótima

### Questão 15

Considere a seguinte situação: Dados 5 palitos, cada jogador pode retirar 1, 2 ou 3 por turno. Perde o jogador que retira o último palito. Utilize a busca MINIMAX para verificar se MAX pode ganhar o jogo.

**Max remove 1 palito:**

**4 palitos restantes**

**Min remove 1 palito:**

**3 palitos restantes**

**Max remove 1 palito:**

**2 palitos restantes**

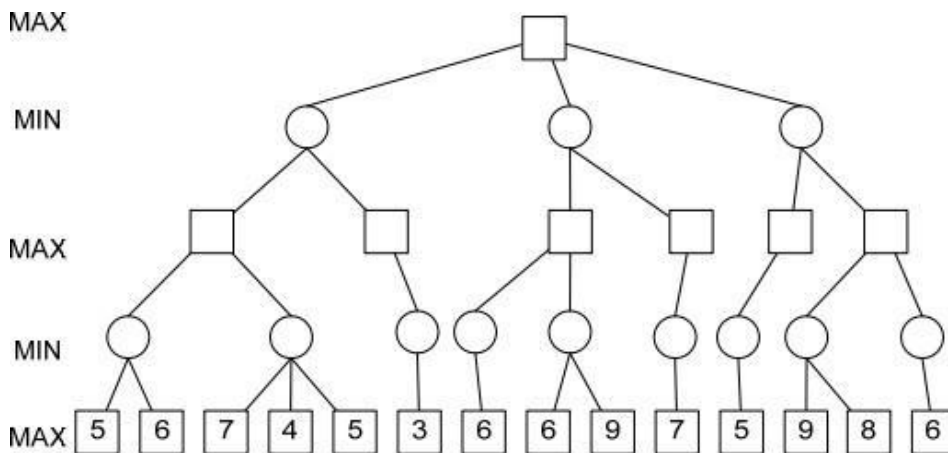
**Max remove 2 palitos:**

**1 palito restante, é vez do MIN, vitória de MAX**

**r: É possível que MAX GANHE.**

**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.

- a) 5
- b) 8
- c) 9
- d) 10
- e) 11