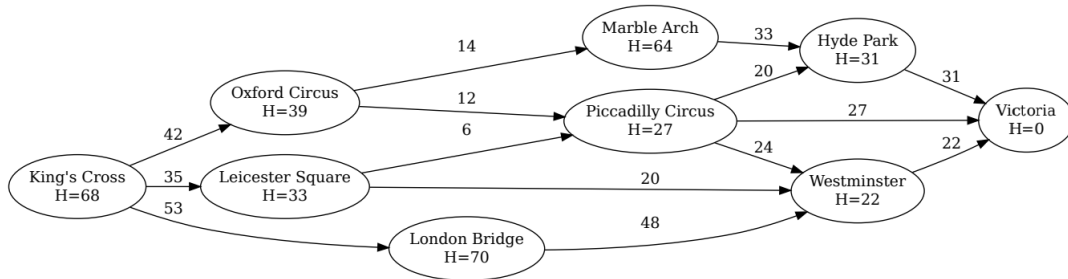


Lista de Exercícios de Busca

Nome: _____

1. Dado o grafo abaixo representando as estações de metrô de Londres, utilize algoritmos de **busca em grafo** para encontrar caminhos partindo da estação **King's Cross** até a estação **Victoria**. O custo das transições está anotado nos arcos do grafo e a estimativa heurística H está anotada nos nodos do grafo.



Responda o que segue, e atente para a direcionalidade do grafo.

- (a) Se utilizarmos Busca em Amplitude¹, qual o **caminho resultante**? Insira na fila os nodos vizinhos de cima para baixo conforme na figura.

Solution: King's Cross → Leicester Square → Piccadilly Circus → Victoria

It.	Closed list	Frontier queue
init	{}	{KC}
1	{KC}	{OC, LS, LB}
2	{KC, OC}	{LS, LB, MA, PC}
3	{KC, OC, LS}	{LB, MA, PC, W}
4	{KC, OC, LS, LB}	{MA, PC, W}
5	{KC, OC, LS, LB, MA}	{PC, W, HP}
6	{KC, OC, LS, LB, MA, PC}	{W, HP, V}
7	{KC, OC, LS, LB, MA, PC, W}	{HP, V}
8	{KC, OC, LS, LB, MA, PC, W, HP}	{V}

- (b) Se utilizarmos Busca de Custo Uniforme², qual o **caminho resultante**?

Solution: King's Cross → Leicester Square → Piccadilly Circus → Victoria

It.	Closed list	Frontier queue
init	{}	{KC}
1	{KC}	{LS (35), OC (42), LB (53)}
2	{KC, LS}	{PC (41), OC (42), LB (53), W (55)}
3	{KC, LS, PC}	{OC (42), LB (53), W (55), HP (61), V (68)}
4	{KC, LS, PC, OC}	{LB (53), W (55), MA (56), HP (61), V (68)}
5	{KC, LS, PC, OC, LB}	{W (55), MA (56), HP (61), V (68)}
6	{KC, LS, PC, OC, LB, W}	{MA (56), HP (61), V (68)}
7	{KC, LS, PC, OC, LB, W, MA}	{HP (61), V (68)}
8	{KC, LS, PC, OC, LB, W, MA, HP}	{V (68)}

- (c) Se utilizarmos Busca Gulosa pela Melhor Escolha³, qual o **caminho resultante**?

¹Breadth-First Search

²Uniform Cost Search

³Greedy Best-First Search

Solution: King's Cross → Leicester Square → Westminster → Victoria

It.	Closed list	Frontier queue
init	{}	{KC}
1	{KC}	{LS (33), OC (39), LB (70)}
2	{KC, LS}	{W (22), PC (27), OC (39), LB (70)}
3	{KC, LS, W}	{V (0), PC (27), OC (39), LB (70)}

(d) Se utilizarmos Busca A^* , qual o **caminho resultante**?

Solution: King's Cross → Leicester Square → Piccadilly Circus → Victoria

It.	Closed list	Frontier queue
init	{}	{KC}
1	{KC}	{LS (68), OC (81), LB (123)}
2	{KC, LS}	{PC (33), W (42), OC (81), LB (123)}
3	{KC, LS, PC}	{V (27), W (42), OC (81), LB (123)}

2. Cite 2 vantagens e 2 desvantagens de usar busca informada:

(a) Vantagens:

Solution:

- Estados promissores recebem prioridade, logo há um melhor uso dos recursos computacionais (tempo X espaço)
- Capaz de tirar vantagem de conhecimento de domínio

(b) Desvantagens:

Solution:

- Necessário ter uma função heurística para o domínio/problema;
- Algoritmo mais complexo.

3. O que os elementos na lista aberta ⁴ representam em algoritmos de busca?

Solution: Os elementos da lista aberta representam os nodos que o algoritmo considera relevante para encontrar a solução.

4. O que é um vizinho do estado atual?

Solution: Um vizinho do estado atual representa um estado possível de alcançar ao aplicar uma ação ao estado atual.

5. A principal vantagem de busca em profundidade é não alocar tantos recursos de memória quanto busca em amplitude/largura. Logo não existe uma tabela com os nodos já visitados. Olhando o

⁴as vezes chamados de elementos de fronteira ou franja

mapa da Figura 1, onde o agente é o ponto verde e o objetivo o ponto vermelho, quais problemas poderiam acontecer para busca em profundidade?



Figura 1: Mapa 1.

Solution: Não encontrar solução e entrar em laço infinito, já que o agente pode sempre explorar o mapa em círculos e nunca gerar backtracking para tentar o caminho em direção ao objetivo.

6. Qual a vantagem de usar busca em profundidade iterativamente aprofundante ⁵?

Solution: A vantagem de usar IDDFS é poder resolver problemas com laços infinitos da busca em profundidade, prevenindo o algoritmo de tentar apenas um caminho e ficar preso neste. IDDFS vai ser limitado por profundidade e sempre retornar.

7. O que acontece com uma busca gulosa (greedy-search) no mapa da Figura 2? Considere que o agente é o ponto verde e o objetivo o ponto vermelho no mapa.

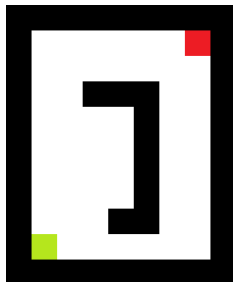


Figura 2: Mapa 2.

Solution: A busca dá prioridade a nodos em linha reta ao objetivo, encontra obstáculos no canto superior e começa a explorar os tiles abaixo e a esquerda até conseguir passar pelo caminho superior até o objetivo.

8. O que acontece quando temos uma função heurística que retorna um valor muito próximo do exato? E quando essa função sempre retorna o mesmo número, 1 por exemplo?

⁵Iterative deepening depth-first search

Solution: Com valores muito próximos do exato não exploramos mais estados do que o necessário, tendo apenas de lidar com o alto custo computacional para obter tal exatidão. Quando sempre é retornado 1 não há como saber se estamos nos aproximando ou não, levando a exploração de vários estados que se afastam da solução. Por outro lado a função é extremamente simples, retornando uma constante.