

Questão 1. Forneça duas representações para cada um dos grafos da Fig. 1, uma utilizando listas de adjacências e a outra uma matriz de adjacências.

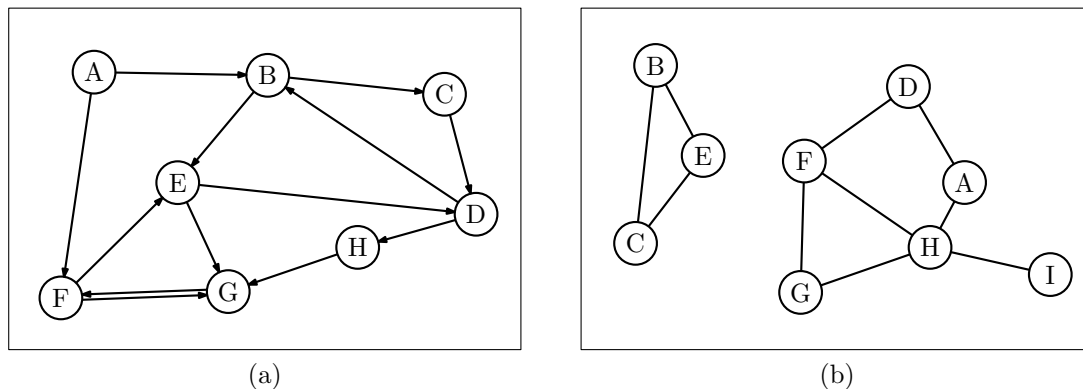


Figura 1: Dois exemplos de grafo, um (a) orientado e o outro (b) não-orientado. Assuma que, em qualquer representação por lista de adjacências, os vizinhos de cada vértice são ordenados segundo a ordem alfabética.

Questão 2. Dado um dígrafo $G = (V, E)$ representado por lista de adjacências, qual o tempo necessário para calcular o grau de saída de todo vértice em G ? E para calcular os graus de entrada?

Questão 3. O grafo *transposto* G^T de um dígrafo $G = (V, E)$ é o par ordenado (V, E') , tal que $E' = \{(v, u) : (u, v) \in E\}$. Descreva dois algoritmos para calcular o grafo transposto de dígrafos, um para quando o grafo estiver representado por lista de adjacências e outro para quando utilizarmos uma matriz de adjacências. Forneça a complexidade de tempo de cada algoritmo.

Questão 4. Realize uma busca em largura nos grafos da Fig. 1 e mostre os valores das distâncias calculadas para cada vértice. Desenhe a árvore de busca em largura resultante.

Questão 5. Para cada célula (i, j) das tabelas abaixo, indique se em qualquer instante durante uma busca em profundidade, pode existir uma aresta ligando um vértice com a cor da linha i a um vértice com a cor da coluna j . Sempre que a existência de uma aresta for constatada, indique quais os tipos de aresta que esta poderia assumir. Realize este procedimento segundo o tipo de grafo indicado na tabela.

.			

(a) Orientado

.			

(b) Não-orientado

Questão 6. Realize uma busca em profundidade nos grafos da Fig. 1 e mostre os valores dos tempos de descoberta e de término para cada vértice. Logo em seguida, classifique as arestas segundo seu tipo: *árvore*, *direta*, *retorno* ou *cruzada*.

Questão 7. Elabore um algoritmo não-recursivo para a DFS utilizando uma pilha.

Questão 8. Modifique a função DFS de tal modo que esta imprima toda aresta do grafo acompanhada de seu tipo. Considere grafos orientados e não-orientados separadamente.

Questão 9. Desenvolva um algoritmo de tempo $O(|V|)$ para verificar se um grafo não-orientado contém um ciclo.

Questão 10. Realize uma ordenação topológica no seguinte DAG:

