

**Universidade Federal de Minas Gerais**  
**Departamento de Computação**  
**Projeto e Análise de Algoritmos – 2024.2**  
**Professor: Marcio Costa Santos**  
**Lista 3**

**Exercício 1.** *Mostre que dado um grafo não orientado  $G = (V, E)$  temos que*

$$\sum_{v \in V(G)} d(v) = 2|E(G)|$$

**Exercício 2.** *Prove que uma árvore com exatamente dois vértices de grau 1 é um caminho.*

**Exercício 3.** *Prove que uma árvore  $T = (V, E)$  com  $\Delta(T) \geq k$  possui pelo menos  $k$  vértices.*

**Exercício 4.** *Definimos o complemento de um grafo simples  $G = (V, E)$  não orientado, como o grafo  $\bar{G} = (V, \bar{E})$  onde*

- $V(\bar{G}) = V(G)$
- $E(\bar{G}) = \{uv \mid uv \notin E(G), u \neq v\}$

*Apresente um algoritmo para essa operação e calcule a sua complexidade em cada uma das duas representações estudadas de um grafo.*

**Exercício 5.** *Definimos a transposição de um grafo simples  $G = (V, A)$  orientado, como o grafo  $G^T = (V, A^T)$  onde*

- $V(G^T) = V(G)$
- $E(G^T) =$  aos arcos de  $G$  com sentido trocado

*Apresente um algoritmo para essa operação e calcule a sua complexidade em cada uma das duas representações estudadas de um grafo.*

**Exercício 6.** *Definimos o quadrado de um grafo simples  $G = (V, A)$  não orientado, como o grafo  $G^2 = (V, A^2)$  onde*

- $V(G^2) = V(G)$
- $E(G^2) = \{uv \mid \exists z \in V(G), z \neq u \neq v, uz \in E(G), vz \in E(G)\}$

*Apresente um algoritmo para essa operação e calcule a sua complexidade em cada uma das duas representações estudadas de um grafo.*

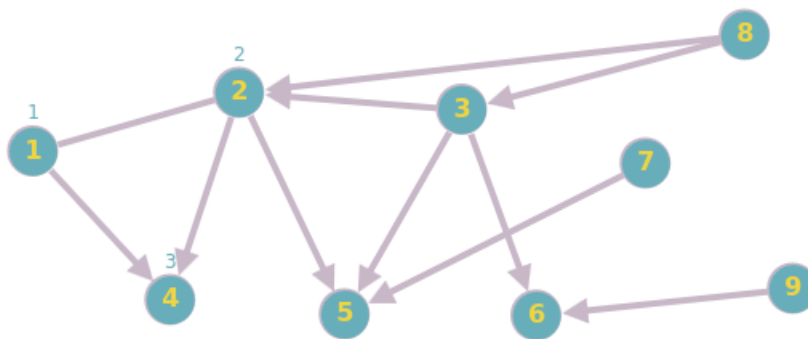
**Exercício 7.** Descreva como podemos modificar as duas estruturas de representação de grafo para acomodar grafos ponderados (grafos que possuem valores numéricos associados com suas arestas ou arcos).

**Exercício 8.** Mostre que uma aresta  $uv$  é

- i. uma aresta de árvore ou de avanço se e somente se  $i[u] < i[v] < f[v] < f[u]$ ;
- ii. uma aresta de retorno se e somente se  $i[v] \leq i[u] < f[u] \leq f[v]$ ; e
- iii. uma aresta de passagem se e somente se  $i[v] < f[v] < i[u] < f[u]$ .

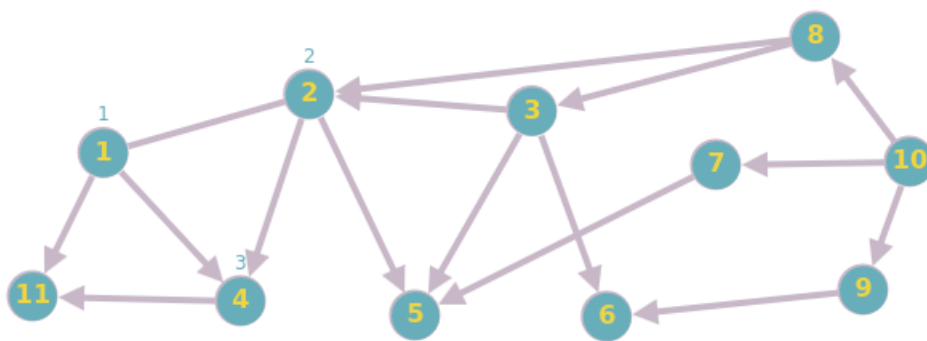
**Exercício 9.** Na descrição da busca em profundidade vista em sala classificamos as arestas do grafo de entrada de acordo após a execução da busca em profundidade. É possível realizar essa classificação durante a execução da busca? Como?

**Exercício 10.** Aplique o algoritmo de busca em profundidade para o grafo abaixo:



**Exercício 11.** Apresente uma versão não recursiva para o algoritmo de busca em profundidade que possua a mesma complexidade da versão recursiva.

**Exercício 12.** Aplique o algoritmo de ordenação topológica no DAG abaixo:



**Exercício 13.** O que acontece quando utilizamos o algoritmo de ordenação topológica em um grafo orientado que contém ciclos? A ordenação obtida faz sentido?

**Exercício 14.** Apresente um algoritmo que determina se um grafo não direcionado possui ou não um ciclo. Qual a complexidade desse algoritmo?

**Exercício 15.** É possível fazer o algoritmo da questão anterior executar em  $O(|V|)$  para toda entrada? Se sim, como?

**Exercício 16.** Como o número de componentes fortemente conexas de um grafo pode mudar pela adição de um arco?

**Exercício 17.** Aplique o algoritmo para determinar componentes fortemente conexas no grafo orientado abaixo:

