

**AULA 14 – O TIPO ABSTRATO DE DADOS GRAPH – PARTE II**

**\*\*\* Para os dois guiões (13 e 14) sobre grafos, entregue num ficheiro ZIP todos os ficheiros de implementação em que completou código \*\*\***

Na aula teórica foram apresentados algoritmos para, dado um grafo orientado, determinar uma **ordenação topológica dos seus vértices, caso exista**.

Neste guião deverão ser implementados e testados os **três algoritmos apresentados na aula teórica (08/06/2021)** que permitem obter uma ordenação topológica dos vértices de um grafo orientado:

- Algoritmo que usa uma **cópia do grafo orientado** e efetua sucessivos **apagamentos dos arcos emergentes de vértices que não tenham arcos incidentes**.
- Algoritmo que não necessita de uma cópia do grafo dado e usa um **array auxiliar** (um dos campos do registo) para sucessivamente **procurar o próximo vértice** a juntar à ordenação topológica.
- Algoritmo que usa uma **fila** para manter o **conjunto dos vértices** que irão ser **sucessivamente adicionados** à ordenação topológica.

**TAREFAS**

- **Completar as funções e testar** os algoritmos pedidos – módulo **GraphTopologicalSorting**.
- Assegurar que os algoritmos executam corretamente para os dois grafos orientados (muito simples) do **exemplo3.c**.
- Testar os algoritmos com grafos orientados mais complexos.

**Atenção:**

Os vértices de um grafo estão sequencialmente numerados: 0, 1, 2, ...

Deve respeitar os protótipos das funções definidos nos vários ficheiros cabeçalho.

Pode criar funções auxiliares (static) locais ao módulo **GraphTopologicalSorting**.