Data Structures and Algorithms. **Bachelor in Computer Science and Engineering** **Bachelor in Data Science and Engineering** **Bachelor in Applied Mathematics and Computing** **Carlos III University of Madrid** **Lab Case. Academic Year 2021-2022**	uc3m | Universidad Carlos III de Madrid

CC BY NC SA

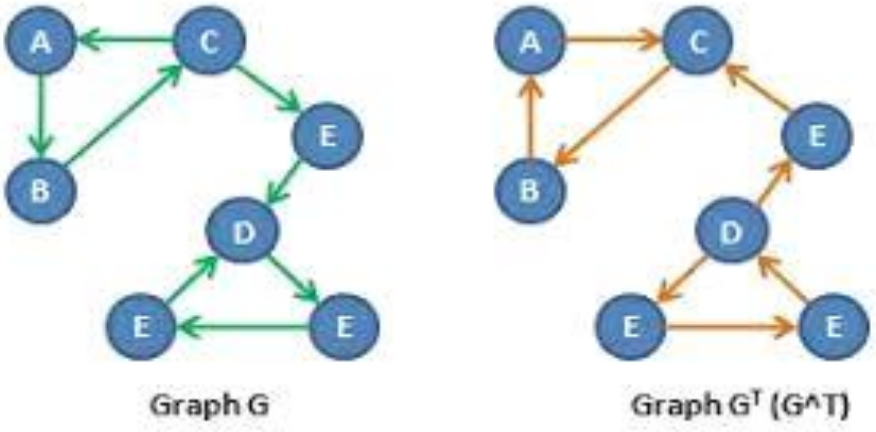
Lab Case Study

The case study consists of three phases. This document describes the statement of the third phase.

Phase 3 - Non-linear ADT data structures (Networks)

Given the class **Graph2**, a child of the class **Graph**, implement the following functions:

- **min_number_edges**: receives a pair of vertices, start and end, and returns the minimum number of edges between both pairs of vertices.
- **transpose**(): returns the transposed graph of the calling graph (self). A graph transposed from a graph G, has the same set of vertices and the same edges, but with the orientation of the edges reversed. That is, if in the graph G, there is an edge (u, v) (i.e. edge with origin u, and destination v), in the transposed graph, the edge will be (v, u), (i.e. origin v, destination u).

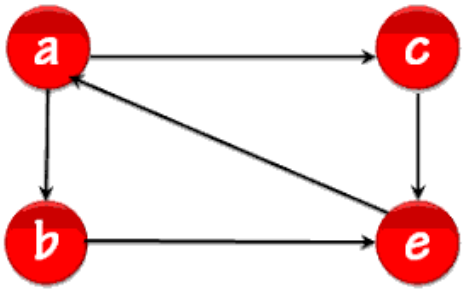


Graph G and its transpose G^T

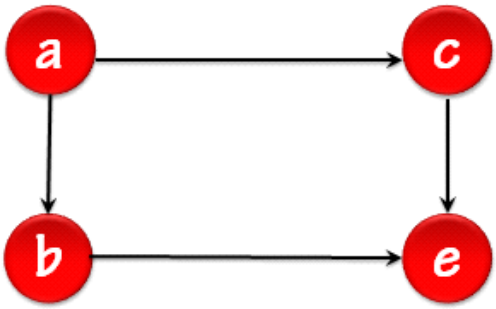
- **is_strongly_connected**(): returns True if the (directed) graph is strongly connected, and False otherwise. Recall that a directed graph is strongly connected if there is a path from any vertex to any other vertex.

For example, the graph in the figure below is a strongly connected directed graph, because it is always possible to find a path from any vertex to any other vertex.

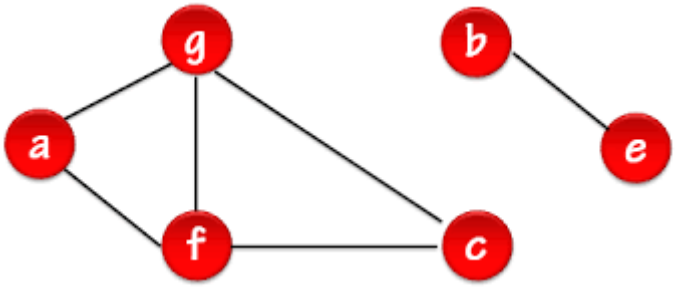
Strongly connected graph:



However, the next directed graph is not strongly connected, because it is not possible to find a path from c to b, for example. In fact, it is not possible to find any path starting from e.



In the case of undirected graphs, the function will return True if the graph is connected and False otherwise. For example, for the following undirected graph, the function should return False because it is an undirected graph:



However, if we only consider the subgraph formed by the vertices a, f, c and g, the function will return True.

Rules:

- A solution is considered correct if it is robust (it has no errors for any input), correct (it solves the problem) and efficient (there can be several solutions, you should always try to find the most efficient solution).
- In this phase, it is allowed to use Python structures such as Lists, Queues or Dictionaries, etc.
- The case study must be carried out by a group of two members (both must belong to the same small group). In no case will groups with more than two members be allowed. Individual groups are not allowed either, as one of the competences to be assessed will be teamwork. If you do not have a partner, please send an email to your internship teacher.
- Mode of delivery: In the small global classroom group, an assignment entitled '**Delivery Phase 2 and 3**' will be posted. The submission date for this third phase will be **16 May, 9.00 am**.
- Delivery format: a zip whose name consists of the two NIAS of the students in the group, separated by a hyphen. For example, *10001234-10005678.zip.* Only one of the two members should upload the group's solution to the assignment. The zip should contain all the skeleton files (with the phase3.py file modified with your solution).
- **Defence**: **16 May 2022 (groups 96, 97, 121)**, and **17 May 2022 (groups 87, 88, 89)** The group will be scheduled at a specific time. The defence of the case study is an oral examination. Attendance is compulsory. If a student does not attend, his/her grade in the case study will be NP. During this defence, the teacher will ask each member of the team a series of questions to be answered individually. Each team member must be able to discuss any of the decisions taken in the design and implementation of any of the functionalities described in the case study. The final grade of the case study will be conditioned by the grade obtained in the defence. If a student is not able to discuss and defend certain decisions, he/she will not be graded on these, even if they have been correctly implemented.
- It is recommended to follow the recommendations described in Zen of Python (https://www.python.org/dev/peps/pep-0020/) and the style guide (https://www.python.org/dev/peps/pep-0008/) published on the official Python website.