



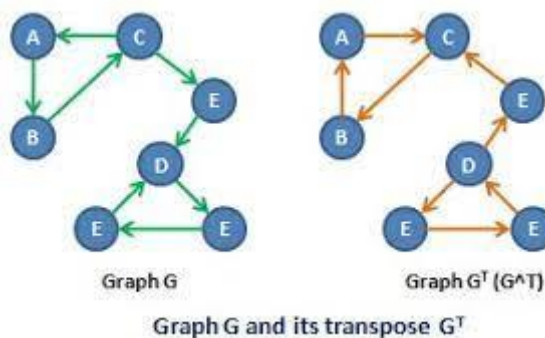
Caso Práctico

El caso práctico se compone de tres fases. Este documento describe el enunciado de la tercera fase.

Fase 3 - Grafos

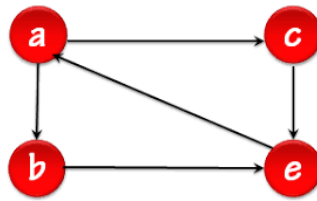
Dada la clase Graph2, hija de la clase Graph, implementa la siguientes funciones:

- **min_number_edges**: que reciba un par de vértices, start y end, y que devuelva el número mínimo de aristas entre ambos pares de vértices.
- **transpose()**: que devuelva el grafo transpuesto del grafo invocante (self). Un grafo transpuesto de un grafo G , tiene el mismo conjunto de vértices y las mismas aristas, pero con la orientación de las aristas invertidas. Es decir, si en el grafo G , existe una arista (u, v) (es decir, arista con origen u , y destino v), en el grafo transpuesto, la arista será (v, u) , (es decir, origen v , destino u).

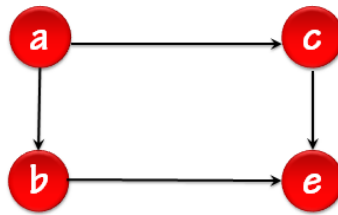


- **is_strongly_connected()**: devuelve True si el grafo (dirigido) es **fuertemente conexo**, y False en otro caso. Recuerda que un grafo dirigido es fuertemente conexo si existe un camino desde cualquier vértice a cualquier otro vértice. Por ejemplo, el grafo de la siguiente figura es un grafo dirigido fuertemente conexo, porque siempre es posible encontrar un camino para ir desde cualquier vértice a cualquier otro vértice.

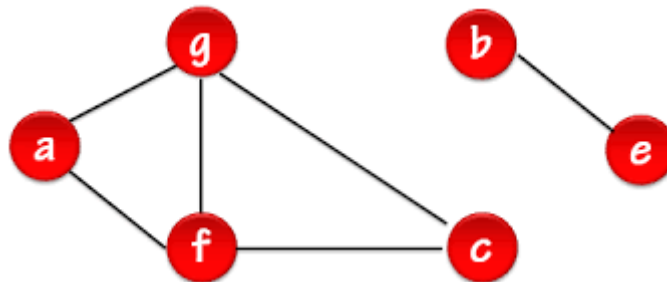
Grafo fuertemente conexo:



Sin embargo, este otro grafo dirigido no está fuertemente conectado, porque por ejemplo, no es posible encontrar un camino desde c a b. De hecho, no es posible encontrar ningún camino que parta de e.



En el caso de grafos no dirigidos, la función devolverá True si el grafo es conexo y False en otro caso. Por ejemplo, para el siguiente grafo no dirigido, la función deberá devolver False porque es un grafo no conexo:



Sin embargo, si únicamente consideramos el subgrafo formado por los vértices a, f, c y g, la función devolverá True.

Normas:

1. Una solución se considera correcta si es robusta (no tiene errores para ninguna entrada), es correcta (resuelve el problema) y es eficiente (pueden existir varias soluciones, debes tratar de buscar siempre la solución más eficiente)
2. En esta fase, si está permitido utilizar estructuras de Python como Listas, Queues o Diccionarios, etc.
3. El caso práctico debe ser realizado por un grupo formado por dos miembros (ambos deben pertenecer al mismo grupo reducido). En ningún caso se permitirá grupos con más de dos miembros. Tampoco se permiten grupos individuales ya que una de las competencias a evaluar será el trabajo en equipo. Si no tienes compañero, por favor, envía un correo a tu profesor de prácticas.

4. Modo de entrega: En el grupo reducido de aula global, se publicará una tarea titulada '**Entrega Fase 2 y 3**'. La fecha de Entrega para esta primera fase será el **16 de mayo, 9.00 am**.
5. Formato de entrega: un zip cuyo nombre está formado por los dos NIAS de los estudiantes que forman el grupo, separados por un guión. Por ejemplo, *10001234-10005678.zip*. Únicamente uno de los dos miembros, será el encargado de subir la solución del grupo a la tarea. El zip únicamente deberá contener los ficheros *phase2.py* y *phase3.py*. No es necesario incluir ningún fichero más.
6. **Defensa: 16 mayo de 2022.** Se citará al grupo a una determinada hora. La defensa del caso práctico es un examen oral. La asistencia es obligatoria. Si algún alumno no asiste, su calificación en el caso práctico será NP. Durante esta defensa, el profesor planteará a cada miembro del equipo una serie de preguntas que deberá responder de forma individual. Cada miembro del equipo debe ser capaz de discutir cualquiera de las decisiones tomadas en el diseño e implementación de cualquiera de las funcionalidades descritas en el caso práctico. La nota final del caso práctico estará condicionada por la nota obtenida en la defensa. Si un alumno no es capaz de discutir y defender ciertas decisiones, no será calificado con respecto a estas, aunque hayan sido correctamente implementadas.
7. Se recomienda seguir las recomendaciones descritas en Zen of Python (<https://www.python.org/dev/peps/pep-0020/>) y la guía de estilo (<https://www.python.org/dev/peps/pep-0008/>) publicada en la página oficial de Python.