Lab02: Circuitos en grafos

1. **Circuito Euleriano.**

   Un circuito Euleriano en un grafo conexo es un camino cerrado, en el que los nodos inicial y final son el mismo, que recorre cada arista del grafo una sola vez.

   En este ejercicio dispones de un fichero con código en Python (`lab02_euler.py`) con partes ya programadas y partes que no lo están. El programa principal tiene una única línea de código, que es una llamada a la función `test`, donde se definen una serie de grafos. Para cada uno de ellos, mediante la instrucción `assert`, se evalúa si tiene un circuito Euleriano o no.

   Este esquema de llamada al método `test` se repetirá a lo largo del curso.

   (a) Programa la función `graph_has_Eulerian _circuit` que, dado un grafo de entrada, devuelve True (cierto) si en dicho grafo hay un circuito Euleriano y False (falso) si no lo hay. La función tiene que decidir sin construir el circuito Euleriano.

   (b) ¿Cuál es el coste de tu algoritmo?

2. **Circuito Hamiltoniano**

   Un camino Hamiltoniano en un grafo conexo es un camino que recorre todos los nodos de un grafo sin pasar dos veces por el mismo nodo. El requisito para que sea un circuito Hamiltoniano es que comience y acabe en el mismo nodo.

   En este ejercicio dispones del fichero (`lab02_hamiltonian.py`).

   (a) Siguiendo el esquema de llamada al método test, programa la función `graph_has_Hamiltonian _circuit` que, dado un grafo de entrada, devuelve True (cierto) si en dicho grafo hay un circuito Hamiltoniano y False (falso) si no lo hay.

   (b) ¿Cuál es el coste de tu algoritmo?

3. **Visualización de grafos.**

   Para comprender mejor lo que estás testeando, hemos incluido un módulo con el que puedes dibujar los grafos que se emplean en los ejercicios anteriores (`to_draw`).

   (a) Abre el fichero `lab02_dibujar.py` y mira cómo hemos usado el módulo `to_draw`.

   (b) Crea la matriz de adyacencia de un nuevo grafo no dirigido y visualiza el grafo.