

## Ejercicios de Repaso:

### Grafos

1. Se tiene un grafo no dirigido  $g$  (instancia de [Graph](#)).
  - a) Dar un algoritmo para determinar si el grafo es conexo o no.
  - b) Dar un algoritmo para determinar cuántas componentes conexas tiene el grafo.
  - c) Dar un algoritmo para determinar si el grafo contiene ciclos o no.
    - a) En caso afirmativo indicar el ciclo encontrado.
    - b) Mejor aún, obtener todos los ciclos que contenga el grafo.\*
  - d) Estimar la eficiencia de las soluciones propuestas.
2. La clase [Graph](#) de las bibliotecas del texto no tiene un método para determinar si entre dos vértices  $u, v$  existe una arista.
  - a) Dar una implementación del método `boolean edgeExists(int u, int v)` reutilizando la misma clase del texto con lista de adyacencia en bolsas.
  - b) Qué tan eficiente es el método anterior? Peor caso, considerar grafo denso y no denso.
  - c) Sería posible lograr una implementación que opere en  $O(1)$ ?
2. Resolver los mismos problemas del punto anterior para un grafo dirigido (instancia de [Digraph](#)).
3. Hacer un procedimiento que convierta un [Graph](#) en un [Digraph](#) equivalente.
4. Hacer un procedimiento que convierta un [Graph](#) / [Digraph](#) a su representación por medio de una matriz de adyacencia.
5. En clase se consideró el problema de encontrar la salida de un laberinto. Analizar la solución propuesta en caso que el grafo sea no conexo o que contenga ciclos. Se garantiza que encuentre la salida?

6. En el problema del laberinto se consideró la solución por medio de un recorrido DFS. Plantear la solución utilizando un recorrido BFS. Cuál de las dos es mejor en el peor caso?

7. En el problema del laberinto no se obtuvo el camino desde el punto de origen hasta el punto de final. Modificar el algoritmo para que retorne el camino como la lista de nodos que lo componen.

8. Se quiere enviar un mensaje "Broadcast" a todos los nodos de una red. Proponer un algoritmo que envíe el mensaje a cada nodo una vez (ningún nodo debe recibir dos veces el mismo mensaje).