Guía de trabajos prácticos: Grafos

1. Un grafo dirigido está formado por los vértices V={A,B,C,D,E}, y su matriz de adyacencia, suponiendo que los vértices ocupan los índices del 0 al 4, es:

0	1	1	1	0
1	0	1	0	1
1	1	0	1	1
0	1	1	0	1
0	1	1	0	0

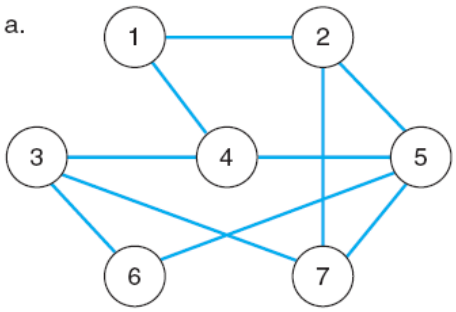
   a) Representar el grafo como un conjunto de nodos y enlaces.
   b) Representar el grafo mediante listas de adyacencias
   c) Dibujar el grafo correspondiente
   d) Mostrar todos los caminos de longitud 2 y longitud 3 que existen
   e) Mostrar su recorrido por profundidad y por anchura comenzando del nodo A.

2. Para los siguientes grafos, muestre el orden en que los nodos serán visitados, comenzando por el nodo 1, siguiendo el algoritmo:

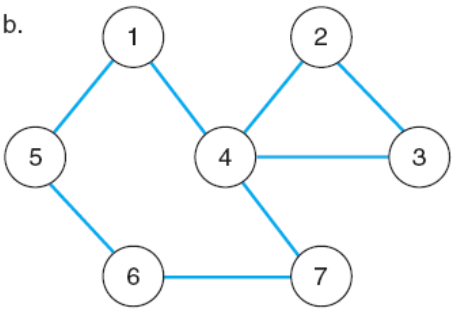
En **profundidad**

En **anchura**

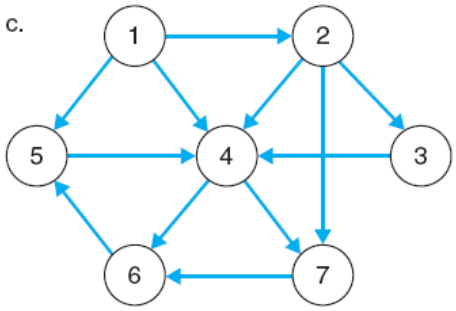
a.



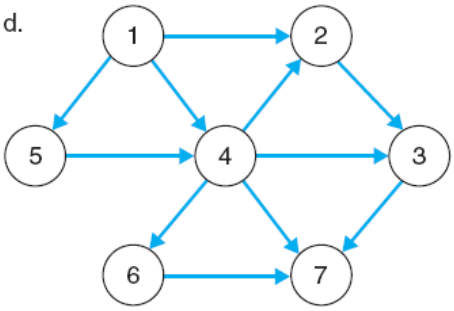
b.



c.



d.



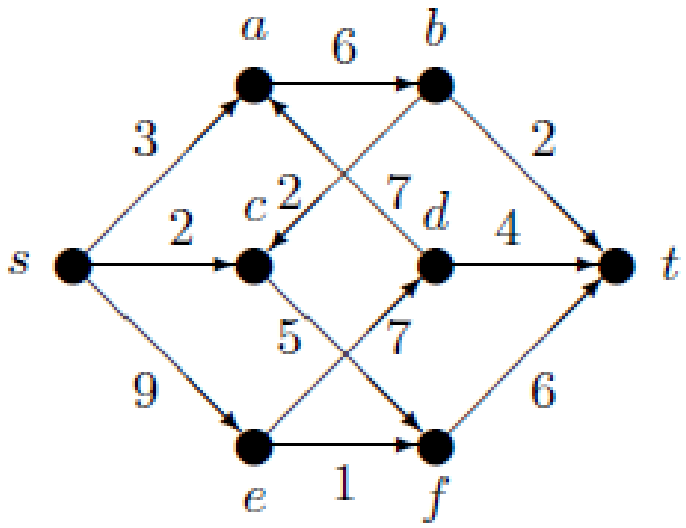
3. Implementar en la clase GrafoDirigido los métodos que permitan determinar el grado positivo y negativo de un nodo.
   3.1. **gradoEntrada**(Nodo unNodo) : integer → Retorna un entero que define el grado positivo del nodo argumento
   3.2. **gradoSalida**(Nodo unNodo) : integer → Retorna un entero que define el grado negativo del nodo argumento.

4. Implementar en la clase GrafoDirigido, un método que dados 2 nodos, determine: si existen un camino entre ambos vértices teniendo en cuenta:
   4.1. Que el método sea recursivo (existeCaminoRecursivo).
   4.2. Que el método sea iterativo (existeCaminoIterativo).
   4.3. Sobrecargar los métodos anteriores de forma tal que se agregue un tercer parámetro N, de tipo numérico, "existeCamino(v1,v2,N)" que retorna verdadero si el camino tiene menos de N saltos
5. Agregue a la clase Grafo los siguientes métodos:
   5.1. **esCompleto**() → retorna true si existe un enlace para cada par de vértices.
   5.2. **ordenTopologico**(): Set<Nodo> → retorna el conjunto de nodos del grafo en orden topológico.

---

6. Considere un conjunto de tareas, t1….tn, que todas deben ser completadas. Suponga que hay restricciones en el orden en el que las tareas pueden ser completadas. En particular, hay una restricción de precedencia, de forma tal que la tarea ti, debe ser realizada que el conjunto de tareas tj...tn.
   6.1. Modele las siguientes tareas como un grafo dirigido con los siguientes elementos: G={t1,t2,t3,t4,t5,t6,t7,78},{(t1,t2),(t1,t3),(t2,t3),(t2,t8),(t3,t6),(t4,t5), (t5,t6),(t5,t7),(t6,t2),(t7,t8)}
   6.2. Es posible calcular una lista de precedencias, de manera tal que si recorremos el grafo, liste TODAS las tareas a realizar, y en el orden que deben ser realizadas?.
   6.3. De ser posible, Muestre la evolución de dicho procedimiento.
   6.4. Si no es posible indique cual es el problema y proponga una solución.

7. Encontrar el flujo máximo que puede ir de s a t en la siguiente red:



Ejercicios complementarios

8. Modifique los métodos del ejercicio 3 para que retornen la lista de **Todos los caminos encontrados entre dos destinos.**

9. Agregar a un grafo, un método llamado **excentricidad** (Vertice unVertice) que retorna un valor entero, que representa la distancia más grande del menor camino que hay entre "unVertice" y cualquier otro vértice del grafo.

10. Dado el siguiente grafo que representa una lista de tareas:
    G={t1,t2,t3,t4,t5,t6,t7,78,t9,t10,t11},{(t1,t4),(t2,t6),(t3,t7),(t6,t4),(t3,t7),(t6,t7),
    (t4,t8),(t6,t8),(t7,t9),(t8,t9),(t3,t10),(t9,t11),(t10,t11)}
    Muestre TODAS las tareas a realizar, y en el orden que deben ser realizadas