# 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 |

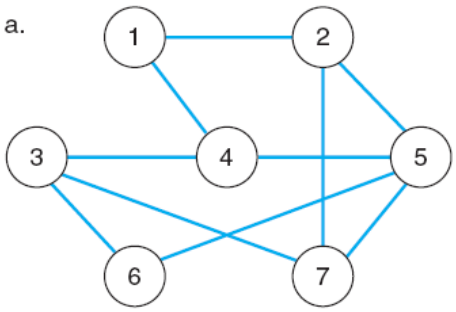
   a) Representar el grafo como un conjunto de nodos y enlaces.
   b) Representar el grafo mediante listas de adyacencias
   c) Dibujar el grafo correspondiente
   d) Mostrar todos los caminos de longitud 2 y longitud 3 que existen
   e) Mostrar su recorrido por profundidad y por anchura comenzando del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

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

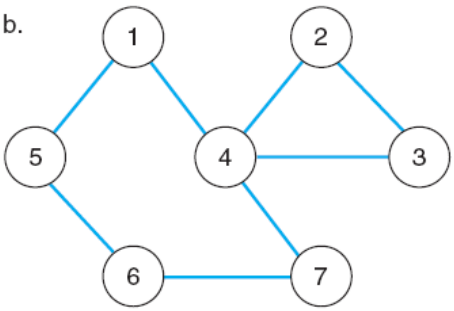
En **profundidad**

En **anchura**

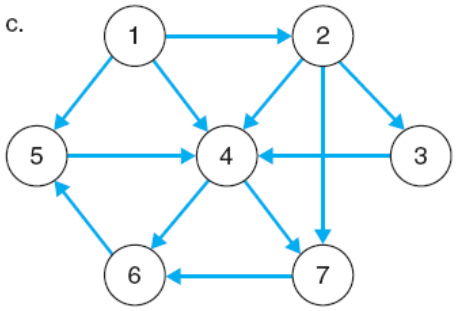
a.



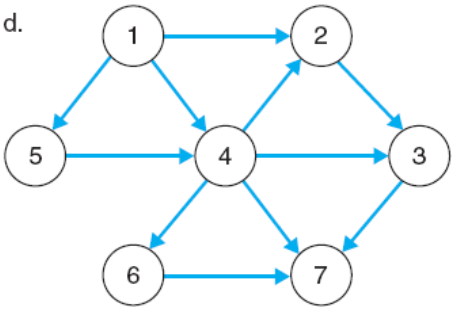

b.




c.




d.



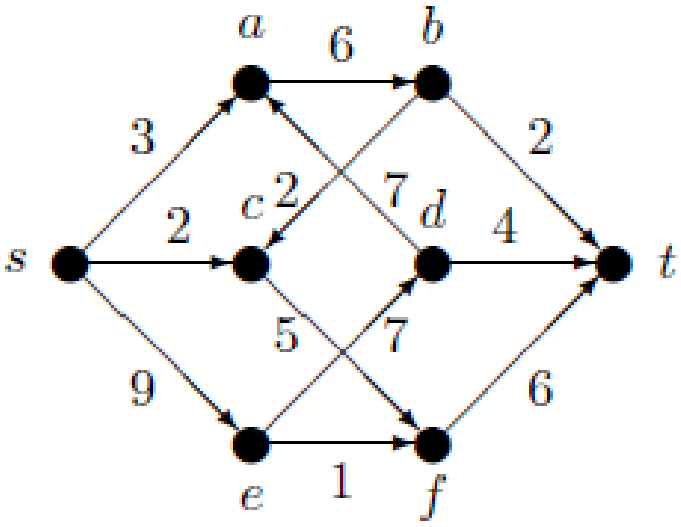
3. Agregue a la clase Grafo los siguientes métodos:
   3.1. **esCompleto**() → retorna true si existe un enlace para cada par de vértices.
   3.2. **excentricidad** (Vertice unVertice) → que retorna un valor entero, que representa la distancia más grande del menor camino que hay entre "unVertice" y cualquier otro vértice del grafo.

4. Implementar en la clase GrafoDirigido, un método que dados 2 nodos, determine: si existen un camino entre ambos vértices teniendo en cuenta que no se pueden realizar más de N saltos.
   4.1. El método a implementar será "existeCamino(v1,v2,N)" que retorna verdadero si el camino tiene menos de N saltos
   4.2. Suponga que los nodos de un grafo tienen como etiqueta un valor entero. Modifique este método para que retorne verdadero si el camino de a lo sumo N Saltos, solo tiene nodos que son pares.
   4.3. Modificar el método 4.2 para que retorne **TODOS** los caminos posibles.
   4.4. Dado un grafo sin pesos, donde los vértices se etiquetan con valores enteros, determine si se cumple la condición de que, dado un vertice par del grafo, es posible alcanzar otro **al menos UN** vértice par, en una cantidad de saltos impar.
   4.5. Dado un grafo con pesos implementar el método existeCamino(v1,v2, **N**, **MAX**) donde se retorna true si es posible alcanzar un camino entre v1 y v2 de **a lo sumo N saltos,** y donde el costo de las aristas recorridas no supere el valor **MAX**

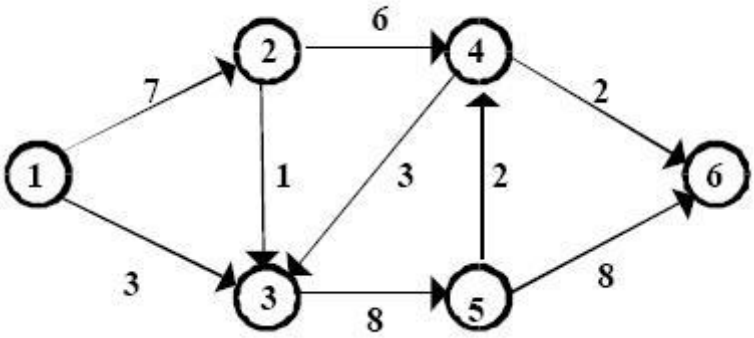
5. Considere un conjunto de tareas, t1….tn, que todas deben ser completadas. Suponga que hay restricciones en el orden en el que las tareas pueden ser completadas. En particular, hay una restricción de precedencia, de forma tal que la tarea ti, debe ser realizada que el conjunto de tareas tj...tn.
   5.1. 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)}
   5.2. 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?.
   5.3. De ser posible, Muestre la evolución de dicho procedimiento.
   5.4. Si no es posible indique cual es el problema y proponga una solución.
6. 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

7. Encontrar el flujo máximo que puede ir de s a t en las siguientes redes:
(*muestre como evoluciona la red en cada iteración)*

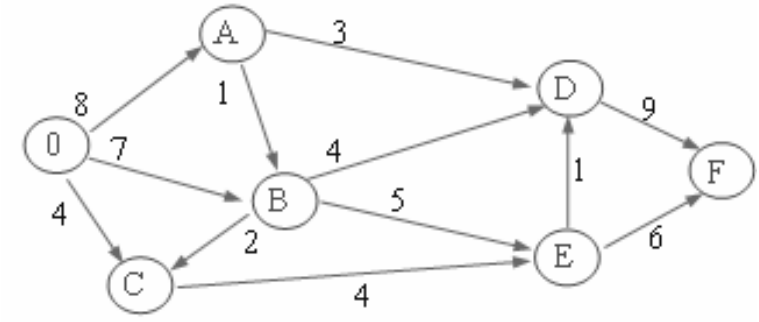
7.1. Red 1




7.2. Red 2




7.3. `



8. Calcular los valores de page rank de los siguientes grafos para cada uno de los nodos asumiendo un factor de amortiguación de 0.25 y 0.75.
   8.1. ¿Cambia en algo la preferencia de nodos con ambos factores?
   8.2. ¿Cambia la magnitud de los valores de page rank?