

PRACTICO DE GRAFOS

MODELADO DE GRAFOS

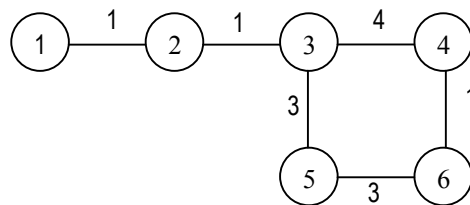
Un grafo es un conjunto de nodos o vértices y un conjunto de arcos dirigidos o no que vinculan a dos vértices que llamamos extremo inicial y final del arco respectivamente.

- 1) Defina clases abstractas Grafo, Vértice y Arco que permitan modelar un grafo.
- 2) Desarrolle una clase GrafoMatrizDeAdyacenciaImp que implemente las operaciones de la clase Grafo del punto anterior. (*)
- 3) Desarrolle una clase GrafoListaDeAdyacenciaImp que implemente las operaciones de la clase Grafo del punto anterior. (*)
- 4) Compare los órdenes de las operaciones para grafos en las implementaciones dadas en 2 y 3 para mejor caso, peor caso, caso promedio.

ALGORITMOS y USO DE GRAFOS

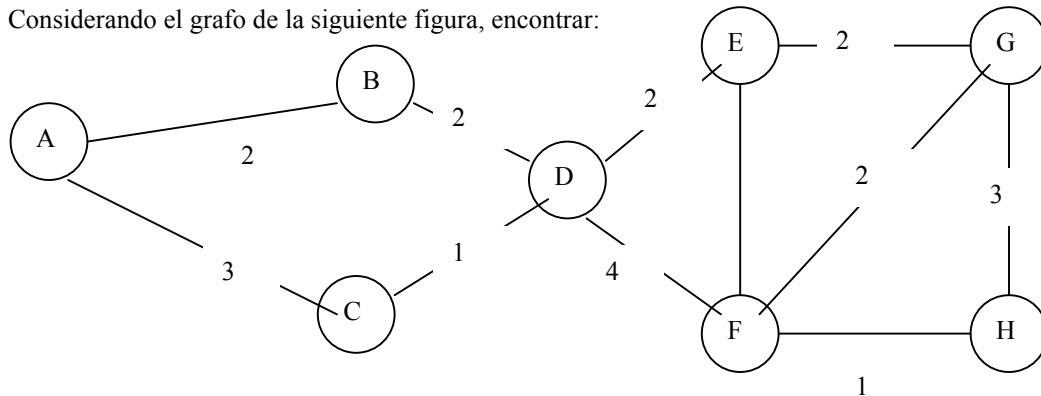
Usando las operaciones para grafos definidas en el apartado anterior resolver:

- 5) Dado un grafo, generar un orden topológico para los vértices.
- 6) Dados un grafo, un vértice origen y otro destino encontrar un camino que partiendo de origen llegue a destino.
- 7) ¿Cómo modificaría el algoritmo anterior para obtener todos los caminos que van de origen a destino?. Implemente el algoritmo.
- 8) Dada una relación y un vértice origen encontrar la longitud del camino más corto **ponderado** que llega hasta cada uno de los demás vértices (Implemente el algoritmo de Dijkstra). (*)
- 9) Desarrolle una rutina que luego de ejecutar el algoritmo de Dijkstra muestre el camino más corto del origen a un vértice v cualquiera. (*)
- 10) En el caso de un grafo acíclico ¿podría ser de utilidad la ordenación topológica de los vértices previa para resolver estos problemas?
- 11) Dado el siguiente Grafo



Indique cuáles son los caminos más cortos con origen en el vértice 1, determinados por el algoritmo de Dijkstra.

12) Considerando el grafo de la siguiente figura, encontrar:



- Un árbol abarcador de costo mínimo con el algoritmo de Prim.
- Un árbol abarcador de costo mínimo con el algoritmo de Kruskal
- Un árbol abarcador en profundidad empezando en A y D

13) Implemente el algoritmo de Prim para la construcción de un árbol de extensión mínimo.

14) Desarrolle los algoritmos de recorrida en profundidad (DFS) y en amplitud de un grafo (BFS).(*)

15) Escriba un programa para encontrar todos los componentes conexos de un grafo no dirigido.

16) Escriba un programa para determinar si un grafo contiene algún ciclo.

17) La clausura transitiva de un Grafo G es otro Grafo G' tal que:

- Los nodos de G' son los nodos de G
- Hay una arista (x,y) en G' si hay un camino entre x e y en G

- Defina un algoritmo que dado un Grafo (representado mediante matrices de adyacencias) determine su clausura transitiva. (*)
- Utilice el algoritmo para:
 - determinar si un Grafo no dirigido es conexo
 - determinar si un Grafo dirigido es fuertemente conexo