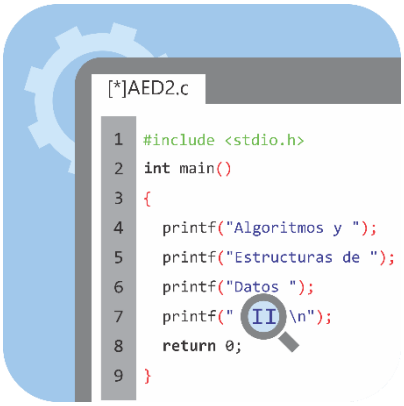
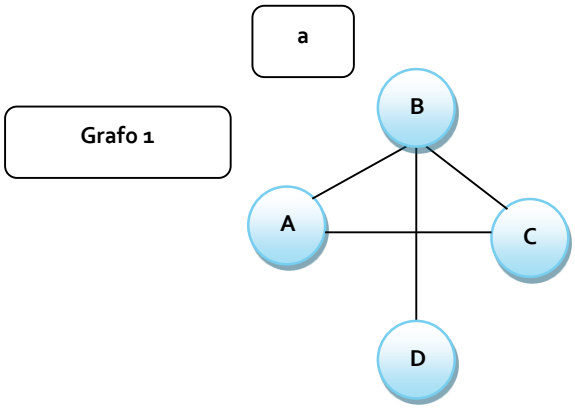


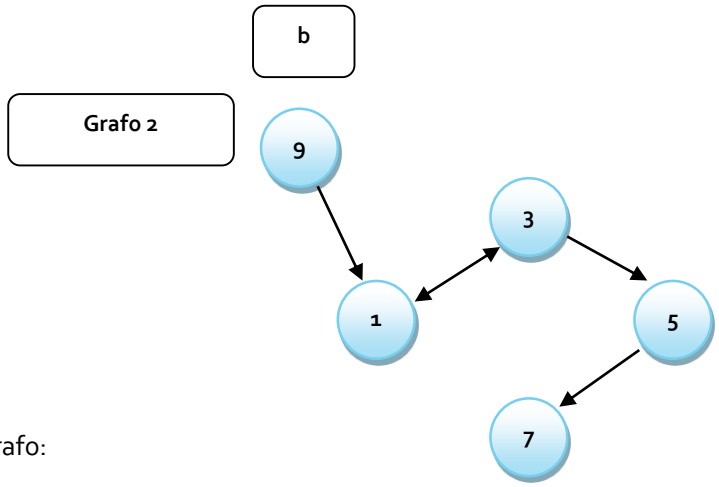
Algoritmos y Estructuras de Datos II

Práctico Nro 9 - Parte 2: Grafos

EJERCICIOS PROPUESTOS

1. Indique los vértices y arcos de los siguientes grafos:





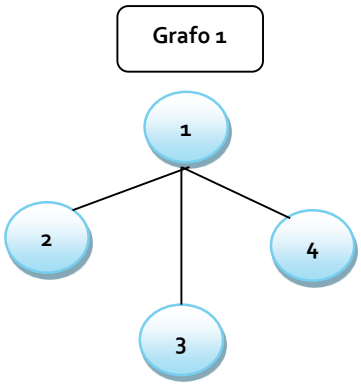
2. A partir de la siguiente información construya el grafo:

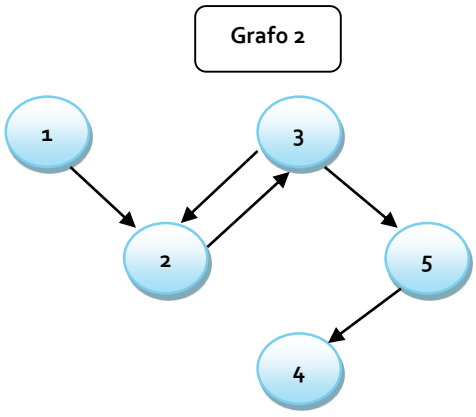
   Vértices V(G) = (X, Y, Z, U, V)

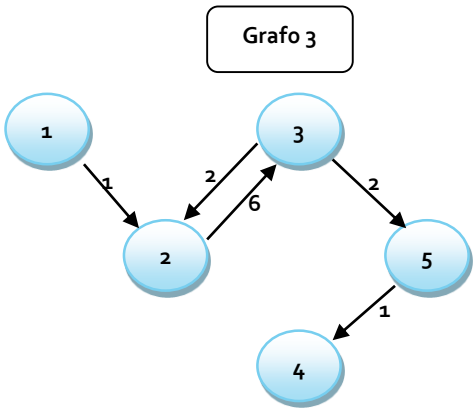
   Aristas A(G1) = (XY, XZ, XV, YZ, YU, ZV, VU)

   Aristas A(G2) = (XY, YX, XZ, XV, YZ, YU, ZV, VZ, UV, VU)

3. Escriba la matriz de adyacencia de los siguientes grafos:







4. A partir de la matriz de adyacencia, reconstruir el grafo:

```
0 1 1 0          0 1 1 1 0
1 0 1 1          1 0 0 1 1
1 1 0 0          1 0 0 1 1
0 1 0 0          1 0 0 0 0
                 0 1 1 0 0
```

5. Escriba una función para iniciar un Grafo
6. Escriba una función para agregar un vértice a un grafo y otro para agregar un arco al grafo.
7. Escriba una función para generar la matriz de adyacencia del ejercicio anterior y otro para visualizarla.
8. Desarrolle un TAD, "Grafo", que permita administrar un grafo ponderado o no ponderado, en donde el valor de las aristas corresponde a un valor entero que, en el caso de ser ponderado recibe el valor del peso, caso contrario 1 ó 0. Considerar que el TAD debe recibir la cantidad de vértices por activar. Funciones: inicializar grado, insertar vértice, eliminar vértice, insertar arco, eliminar arco, mostrar matriz ponderada, mostrar matriz de adyacencia. Luego implemente un programa que utilice el TAD "Grafo".

EJERCICIOS COMPLEMENTARIOS

1. Desarrolle un TAD "ArbolEP" que defina las funciones para el manejo de un árbol binario de búsqueda de números enteros positivos. Funciones: básicas, insertar, eliminar, buscar, mostrar (en inorden).
   Luego implemente un programa que permita probar el TAD "ArbolEP".
2. Desarrolle un TAD, "ArbolReal", que permita administrar un árbol binario de búsqueda de números reales. Funciones: básicas, insertar, eliminar, buscar, mostrar (preorden), devolver la suma de todos los nodos del árbol (recursivo).
3. Sea un grafo ponderado G = (V,A), donde V es su conjunto de vértices, A el conjunto de arcos y sea L[i,j] su matriz de adyacencia. Queremos calcular el camino más corto entre un vértice vi tomado como origen y cada vértice restante vj del grafo. (Alg. Dijkstra)
4. En un plano con varios nodos de coordenadas (x, y). Tu trabajo es como conectar todos los nodos de forma que utilice la menor cantidad de tinta posible (Alg. Kruskal).