

# EjercicioMatrizAdyacencia

## Estructuras de Datos

### Tema 5: grafos

1º Grado en Ingeniería de la Computación

© Profesor Dr. Carlos Grima Izquierdo ([www.carlosgrima.com](http://www.carlosgrima.com))

URJC ([www.urjc.es](http://www.urjc.es))

Programa en C++ la clase "Grafo" (implementada mediante una matriz de adyacencia binaria). Es un grafo dirigido, no acíclico, no multigrafo, pseudografo, no necesariamente conexo, no etiquetado. Tendrá los siguientes métodos públicos (añade todos los privados que consideres necesario):

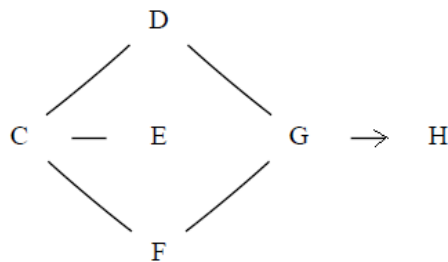
- Constructor (construye un grafo vacío) y destructor.
- Inserta un nodo cuyo nombre se proporciona (el nombre es una sola letra). El nodo se inserta desconectado del resto. Habrá que añadir una columna y fila más a la matriz de adyacencia interna.
- Inserta un arco. Para ello, se da como parámetro un par de nodos que tienen que existir previamente en el grafo. El grafo es dirigido, así pues el orden en que se pasen los dos nodos es relevante. No se puede insertar un arco que ya exista. Ej: si intentamos insertar por segunda vez un arco que vaya de "A" a "B", la precondition del método no nos dejará. Sin embargo si podremos insertar de "B" a "A", pues al ser el grafo dirigido se trata de un arco diferente.
- Elimina un nodo, cuyo nombre se pasa. El nodo debe existir previamente en el grafo. Se elimina automáticamente todos los arcos que partan o que lleguen a ese nodo. No hagas una eliminación "perezosa".
- Elimina un arco, cuyos dos extremos (dos nodos ordenados) se pasa. El arco debe existir previamente en el grafo. Al igual que en insertar, no es lo mismo un arco que va de "A" a "B" que de "B" a "A".
- Imprime el grafo. Para ello, se imprimirá la matriz de adyacencia por pantalla. Se imprimirá tanto el nombre de las columnas, como el nombre de las filas, como lo que contiene cada celda (un 0 ó un 1 únicamente). Todo debe quedar bien alineado en la pantalla (utiliza las capacidades de alineación de printf si es necesario).

Ten en cuenta lo siguiente:

- Lo más lógico es que la matriz de adyacencia esté representada por un array de objetos de tipo ListaContigua (cada objeto de tipo ListaContigua representa a una fila de la matriz). Por lo tanto la clase Grafo tendrá como atributo el puntero al primer elemento de dicho array.
- Reutiliza la clase ListaContigua del EjercicioListaContigua2.
- Como el nombre de cada nodo es una sola letra, utiliza un objeto de tipo "string" (que no es más que una lista de letras) para guardar el conjunto de todos los nodos. Utiliza los métodos de la clase "string" para poder insertar y eliminar nodos del grafo.

Añade al proyecto un "main" que realice la siguiente batería de pruebas:

- Crea el objeto grafo vacío. Lo imprime.
- Inserta, en este orden, los nodos A, B, C, D y E. Imprime el grafo.
- Añade los arcos: (A,A), (A,C), (A,E), (B,B), (B,D), (C,A), (C,C), (C,E), (D,B), (D,D), (E,A), (E,C), (E,E). Imprime el grafo.
- Borra el nodo B. Imprime el grafo.
- Borra el nodo A. Imprime el grafo.
- Borra el nodo E. Imprime el grafo.
- Inserta, en este orden, los nodos H, G, F, E. Imprime el grafo.
- En este punto, los únicos arcos son (C,C) y (D,D). Borra ambos. Imprime el grafo.
- Inserta los siguientes arcos: (C,D), (D,C), (D,G), (G,D), (G,H), (C,E), (E,C), (C,F), (F,C), (F,G), (G,F). Imprime el grafo. El grafo que hemos formado es el siguiente (fíjate en que todos los arcos tienen dos sentidos, excepto de G a H):



```

Consola de depuració...
La matriz esta vacia

  A B C D E
A 0 0 0 0 0
B 0 0 0 0 0
C 0 0 0 0 0
D 0 0 0 0 0
E 0 0 0 0 0

  A B C D E
A 1 0 1 0 1
B 0 1 0 1 0
C 1 0 1 0 1
D 0 1 0 1 0
E 1 0 1 0 1

  A C D E
A 1 1 0 1
C 1 1 0 1
D 0 0 1 0
E 1 1 0 1

  C D E
C 1 0 1
D 0 1 0
E 1 0 1

  C D
C 1 0
D 0 1

```

```
Seleccionar C:\Windows\system32\cmd.exe

  C D H G F E
C 1 0 0 0 0 0
D 0 1 0 0 0 0
H 0 0 0 0 0 0
G 0 0 0 0 0 0
F 0 0 0 0 0 0
E 0 0 0 0 0 0

  C D H G F E
C 0 0 0 0 0 0
D 0 0 0 0 0 0
H 0 0 0 0 0 0
G 0 0 0 0 0 0
F 0 0 0 0 0 0
E 0 0 0 0 0 0

  C D H G F E
C 0 1 0 0 1 1
D 1 0 0 1 0 0
H 0 0 0 0 0 0
G 0 1 1 0 1 0
F 1 0 0 1 0 0
E 1 0 0 0 0 0

Presione una tecla para continuar . . .
```