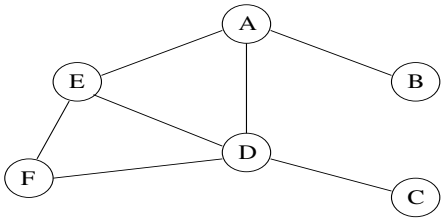
Estructura de datos y de la información
Boletín de problemas - Tema 11

1. Suponer que queremos almacenar la información del grafo de la siguiente figura:



   a) Si utilizamos una implementación basada en una matriz de adyacencia, ¿Qué información guardarán el vector de nodos y la matriz de adyacencia?

   b) Si utilizamos una implementación basada en un vector de listas de adyacencia, ¿Qué información guardarán el vector y las listas asociadas a cada nodo?

2. Implementar la operación `findelem` incluida en el grafo basado en un vector de listas.

3. Implementar la operación de recorrido de un grafo en profundidad suponiendo que el grafo se basa en un vector de listas. ¿Qué problema se plantearía si queremos recorrer un grafo dos veces consecutivas? ¿Cómo solucionarías el problema?

4. Realizar trazas de la operación de recorrido en profundidad comenzando en cada uno de los nodos del grafo del ejercicio 1.

5. Implementar la operación de recorrido en anchura de un grafo suponiendo que el grafo se basa en un vector de listas.

6. Realizar trazas de la operación de recorrido en anchura comenzando en cada uno de los nodos del grafo del ejercicio 1.

7. Implementar la operación `erase_nodo` incluida el grafo basado en un vector de listas.

   a) Cuando se crea un hueco en el vector, ¿accederá a dicho hueco alguna de las operaciones del grafo?

   b) Supón que con el fin de optimizar el uso del vector, cada vez que borramos un nodo, desplazamos todos los situados después. ¿Qué problema se plantearía con las listas de adyacencia asociadas a cada nodo?

   c) Para evitar el problema anterior, supongamos que utilizamos el campo `visitado` de cada nodo para marcar los nodos vacíos. Dicho campo pasará a ser un entero, de modo que si vale 0, el nodo no ha sido visitado en un recorrido, si vale 1, el nodo ha sido ya visitado, y si vale 2, el nodo está vacío. Modificar la operación `BorrarNodo` para tener en cuenta este cambio.

   d) Si es necesario, modificar el resto de operaciones para tener en cuenta el cambio anterior en el campo `visitado`.

8. Implementar el resto de operaciones incluidas en el grafo basado en un vector de listas teniendo en cuenta la gestión de los huecos vista en el ejercicio anterior.

9. Suponer que un grafo se implementa mediante una lista de nodos en lugar de un vector de nodos. Implementar las operaciones de grafo utilizando la lista de la librería STL, tanto para la lista de nodos como para las listas de adyacencia asociadas a cada nodo.

10. Supongamos que trabajamos con un grafo ponderado en el que los arcos tienen un valor numérico. Implementar una operación de la clase grafo que dada la posición de un nodo, devuelva el valor del nodo directamente conectado con él con el arco de mínimo peso.

11. Implementar una operación de la clase grafo que dadas las posiciones de dos nodos: origen y destino, devuelva cierto si están conectados y falso si no lo están. Los nodos pueden estar conectados directamente, o bien a través de otros. La operación deberá escribir por pantalla el camino seguido desde el nodo origen al destino. Para implementarla puedes usar una estrategia de recorrido en profundidad o en anchura.

12. Supongamos que trabajamos con un grafo ponderado en el que los arcos tienen un valor numérico. Modificar la operación del ejercicio anterior para que devuelva la suma de los pesos de los arcos en el camino entre los dos nodos. Si los nodos no están conectados, devolverá 0.

13. Implementar un algoritmo iterativo que recorra el grafo en profundidad.

14. Un grafo modeliza los vuelos de una determinada compañia área. En los nodos se tiene los aeropuertos y hay un arco de A a B si la compañia opera un vuelo entre el aeropuerto A y el aeropuerto B. Dados 2 nombres de aeropuertos, A1 y A2, implementar una operación del grafo que devuelva los aeropuertos en los que hacemos escala para ir de A1 a A2, teniendo en cuenta que queremos hacer el menor número posible de escalas.

15. Modificar el algoritmo de recorrido en profundidad de un grafo para que devuelva la longitud del camino simple más largo existente en el grafo a partir del nodo del cual se inicia el recorrido.