Laboratorio Nro. 1: Implementación de grafos

Juan José Parra
Universidad Eafit
Medellín, Colombia
jjparrad@eafit.edu.co

Luisa María Vásquez
Universidad Eafit
Medellín, Colombia
lmvasquez@eafit.edu.c

3) Simulacro de preguntas de sustentación de Proyectos

3.1) Escriban una explicación entre 3 y 6 líneas de texto del código del numeral 1.1. Digan cómo funciona, cómo está implementado el grafo con matrices y con listas que hizo, destacando las estructuras de datos y algoritmos usados

a) En la implementación con matrices de adyacencia, se crea una matriz de enteros sencilla del tamaño ingresado por el usuario, que sería la cantidad de vértices del grafo, y se inicializan todos los valores en 0. Al ir agregando arcos, se ingresa a la matriz en la fila "Nodo de salida" y a la columna "Nodo de llegada" y se cambia el valor por el peso del arco, de manera que los valores de la fila "n" diferentes a 0 , son los sucesores del nodo "n" del grafo.

b) En la implementación con listas de adyacencia se crea una lista principal que contiene, por cada vértice del grafo, una lista de objetos tipo "Pareja", que representan los sucesores de este; al añadir un arco se ingresa a la lista principal en el índice "nodo de salida" y a su lista de "Parejas", se añade un nuevo objeto creado con el nodo de llegada y el peso del arco.

3.2) ¿En qué grafos es más conveniente utilizar la implementación con matrices de adyacencia y en qué casos en más convenientes listas de adyacencia? ¿Por qué?

En los grafos es conveniente utilizar listas de adyacencia cuando los nodos del grafo sólo se relacionan con algunos y no con todos los nodos del grafo, es mejor las listas debido a que se reduciría el espacio de memoria utilizada, en cambio con matriz se utilizaría memoria no necesaria.

Para el caso contrario, es decir, que todos estén conectados con todos o con la mayoría, es mejor utilizar matriz de adyacencia ya que si se requiere saber si existe un arco o su peso , el peor de los casos es O(1).

3.3) Para representar el mapa de la ciudad de Medellín del ejercicio del numeral 1.4, ¿qué es mejor usar, Matrices de Adyacencia o Listas de Adyacencia? ¿Por qué?

En el caso del mapa de Medellín la mejor opción son las listas de adyacencia, debido a que gastan menos memoria que las matrices de adyacencia, la razón principal de esto es que en el mapa no todos los nodos se relacionan con todos, sino que se relacionan con los vecinos, por lo que una matriz no sería tan útil, ya que son miles de nodos ,es decir una matriz exageradamente grande que gastaría demasiada memoria.

3.4) Teniendo en cuenta lo anterior, respondan: ¿Qué es mejor usar, Matrices de Adyacencia o Listas de Adyacencia? ¿Por qué?

Para el caso de los usuarios de Facebook es mejor utilizar listas de adyacencia ya que las personas de esta red social no son amigos de todo el resto de personas que utilizan Facebook, por lo que al usar listas de adyacencia optimizará la memoria y se agilizarán las operaciones como listar, buscar, guardar, entre otros. Haciendo un cálculo con la información dada donde Facebook tiene alrededor de 100 millones de usuarios y cada usuario tiene en promedio 200 amigos tendríamos que, usando matrices de adyacencia, estaríamos reservando alrededor de 10.000 billones de espacios de memoria para contener toda la información, lo cual es poco eficiente.

3.5) Teniendo en cuenta lo anterior, para representar la tabla de enrutamiento, respondan: ¿Qué es mejor usar, Matrices de Adyacencia o Listas de Adyacencia?

En el caso de los enrutadores, la mejor opción es la matriz de adyacencia, debido a que cada enrutador guarda en la tabla la información que comparte con otro enrutador, lo que hace las cosas mucho más sencillas y eficaces, ya que basta con ubicar la fila que representa un enrutador, y la columna que representa a otro, para hallar información entre ellos. Por otra parte, debido a que los enrutadores de cierto espacio se relacionan todos entre sí sabiendo la distancia el uno con el otro, representarían un grafo en el que cada nodo está conectado con los otros, por lo que no se desperdicia memoria y se podría acceder a la información de manera más eficiente.

3.7) Expliquen con sus propias palabras la estructura de datos que utilizan para resolver los problemas, y cómo funcionan los algoritmos realizados en el numeral 2.1 y los ejercicios opcionales que hayan hecho del punto 2.

En el ejercicio en línea se observó que una de las maneras de determinar si un grafo era "Bicolorable" o no era determinando si el grafo era un grafo bipartito completo, en el cual todos los vértices de un conjunto V1 están conectados con todos los vértices de un conjunto V2 , de manera que el grado de un vértice que pertenece a V1 es la cantidad de vértices pertenecientes a V2.Así , el algoritmo divide los vértices del grafo en dos conjuntos según su grado y verifica esta ultima condición , si es correcta retorna que el grafo es "Bicolorable".

3.7) Calculen la complejidad del ejercicio 2.1.

Complejidad del algoritmo isBicolorable: O(n*m)

3.8) Expliquen con sus palabras las variables (*qué es 'n', qué es 'm', etc.*) del cálculo de complejidad del numeral 3.7

En un cálculo de complejidad las variables significan el número de ejecuciones que se hace por cada una de estas; en este caso la variable "n" es el tamaño de la lista del grafo y "m" es el tamaño del ArrayList usado debido a que hay ciclos que se ejecutan "n" veces y agregan elementos al ArrayList, que en el peor de los casos tiene que hacer una copia de sí mismo para agregar un elemento, lo que se ejecuta "m" veces.

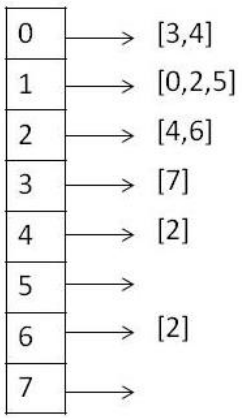
4) Simulacro de Parcial

1)

	0	1	2	3	4	5	6	7
0				1	1			
1	1		1			1		
2					1		1	
3								1
4			1					
5								
6			1					
7								

DOCENTE MAURICIO TORO BERMÚDEZ
Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627
Correo: mtorobe@eafit.edu.co

2)



3) b) $O(n^2)$

DOCENTE MAURICIO TORO BERMÚDEZ
Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627
Correo: mtorobe@eafit.edu.co