

## Laboratorio Nro. 5: Implementación de Grafos

**Carla Daniela Rendón**

Universidad Eafit  
Medellín, Colombia  
cdrendonb@eafit.edu.co

**Sebastián Arboleda Botero**

Universidad Eafit  
Medellín, Colombia  
sarboledab@eafit.edu.co

3.2. Un grafo es una representación simbólica de los elementos constituidos de un sistema o conjunto, mediante esquemas gráficos. En el algoritmo a realizar se nos pide una implementación de grafos con matrices de adyacencia y con lista de adyacencia. Para construir una matriz de adyacencia, se deben tomar en cuenta los siguientes aspectos: El primero es que a cada elemento  $(i,j)$  se suma 1, cuando exista una arista que una los vértices (nodos)  $i$  y  $j$ ; el segundo es que si una arista es un bucle, y el grafo es no dirigido, se suma 2 en vez de 1; y finalmente; cada elemento  $(i,j)$  valdrá 0, cuando no exista una arista que una los nodos  $i$  y  $j$ . Para el caso de una lista de adyacencia esta se suele representar con un array de  $|V|$ . Listas de Vértices:  $G[v]$  es una Lista de los Vértices Adyacentes a  $v$  ( $v \in V$ ). Debemos además tener en cuenta que el tiempo de acceso es de tipo  $O$  (Grado de salida de  $G$ ).

3.3. En el caso de la implementación de grafos con matrices de adyacencia esta es más conveniente para grafos densos (con muchas aristas). Y es más conveniente usar una implementación de lista adyacente en un grafo dirigido ya que se consume menos memoria que con un grafo no dirigido.

3.4. Es más conveniente usar matrices de adyacencia ya que estas son más útiles con grafos densos que posean muchas aristas.

3.5. Depende del algoritmo, en el caso de las matrices con adyacencia se consume mucha memoria, pero es útil para grafos densos. En caso de que se tenga un grafo dirigido, resulta una buena opción usar Listas de adyacencia.

3.6. Considero que es mejor implementar una lista de adyacencia ya que este algoritmo se puede implementar mediante un grafo dirigido.

3.7. La complejidad para el algoritmo planteado es de  $O(n^2)$ .

3.8. Las variables  $n$ ,  $m$  representan las entradas a la función recursiva, es decir, si tiene una sola entrada esta será designada como  $n$ , si tiene dos entradas una será designada como  $n$  y la otra como  $m$ , así aumentará la cantidad de variables dependiendo de las entradas que tenga la función.

**DOCENTE MAURICIO TORO BERMÚDEZ**

Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627

Correo: mtorobe@eafit.edu.co

#### 4) Simulacro de Parcial

1.

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

2.

0 → [3,4]

1 → [0,2,5]

2 → [1,4,6]

3 → [7]

4 → [2]

5 → []

6 → [2]

7 → []

3.B)  $O(n^2)$