

# Práctica 6 - Introducción a grafos

## NOTAS PRELIMINARES

Los objetivos de esta práctica son:

- Ejercitar las definiciones básicas de teoría de grafos.
- Analizar las complejidades de los algoritmos más sencillos sobre grafos dependiendo de la estructura de representación usada.
- Comprender recorridos clásicos sobre grafos, sus diferencias y sus usos.

### Ejercicio 1

Dado un digrafo representado por medio de lista de adyacencias, cuánto cuesta calcular el grado de salida de cada vértice? Y el de entrada?

### Ejercicio 2

Dado un árbol binario completo de 7 vértices, representarlo por medio de lista de adyacencias y por matriz de adyacencias. Considerar a los vértices numerados por niveles.

### Ejercicio 3

El cuadrado de un grafo dirigido  $G = (V, E)$  es el grafo  $G^2 = (V, E^2)$  donde  $(u, w) \in E^2$  si para algún  $v \in V$ ,  $\{(u, v), (v, w)\} \in E$ . Es decir,  $G^2$  contiene un eje entre  $u$  y  $w$  siempre y cuando existe un camino de exactamente dos ejes entre  $u$  y  $w$  en  $G$ . Describir algoritmos eficientes para obtener  $G^2$  a partir de  $G$  para los casos en que este se encuentre representado por matriz de adyacencias o bien listas de adyacencias.

### Ejercicio 4

La mayor parte de los algoritmos sobre grafos requieren una complejidad de al menos  $V^2$  (es decir,  $\Omega V^2$ ) al usar matriz de adyacencias como representación. Mostrar que una excepción a esta regla es el problema de determinar si un digrafo contiene una fuente universal, es decir, uno que es adyacente a todos los demás y ninguno llega a él.

### Ejercicio 5

Supongamos que en lugar de una lista de adyacencias, cada posición del arreglo  $Adj[u]$  contiene un conjunto de adyacentes a  $u$  implementado como una tabla de *hash*. Si consultar por la pertenencia de un eje es igual de probable que hacerlo por cualquier otro, cuál es la complejidad esperada de buscar un eje en esta estructura de datos? En qué tipo de grafos esta estructura alcanza limitaciones? Sugiera un cambio a su estructura para que sea más versátil a distintos tipos de grafos.

### Ejercicio 6

- a. Determinar si un grafo contiene algún ciclo. Cuál es su complejidad?
- b. En términos de grafos, cuál es el objetivo del algoritmo del banquero para evitar *deadlocks* visto en sistemas operativos? Cuál es su complejidad?

### Ejercicio 7

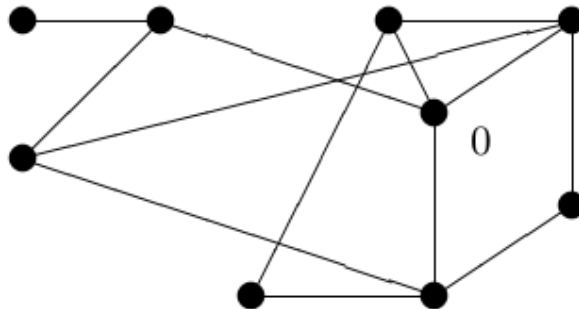
- a. Escribir una función que dado un grafo  $G$  verifique si cada eje de  $G$  pertenece a un circuito simple de  $G$ . Analizar distintas estructuras de datos para representar  $G$ .
- b. Escribir un procedimiento que dado un grafo  $G$ , si puede lo oriente de forma que se convierte en un digrafo fuertemente conexo. Analizar distintas estructuras de datos. Utilizar la función definida en **a**.
- c. Analizar las complejidades de los algoritmos de **a** y **b**.

### Ejercicio 8

- a. Escribir un algoritmo que dado un grafo  $G$  devuelva un subgrafo completo máximo del mismo.
- b. Escribir un algoritmo que dado un grafo  $G$  liste todos sus subgrafos completos maximales.
- c. Analizar las complejidades de los algoritmos de **a** y **b**.

### Ejercicio 9

- Usar BFS para numerar el grafo de la figura a partir del nodo marcado con 0.
- Idem **a** para DFS.



### Ejercicio 10

Escribir un algoritmo recursivo y otro iterativo que implementen BFS. Qué técnicas utilizan estos algoritmos?

### Ejercicio 11

- Probar que con DFS se alcanzan todos los vértices de un grafo conexo.
- Idem **a** para BFS.

### Ejercicio 12

- Escribir un programa para contruir un árbol generador dado por DFS dada la matriz de adyacencia de un grafo.
- Idem para BFS.

### Ejercicio 13

Probar que la altura de cualquier árbol generador determinado por DFS a partir de una raíz dada  $r$ , es mayor o igual que la altura de un árbol generador construido por BFS a partir de la misma raíz  $r$ .

### Ejercicio 14

- Cómo podrá reescribirse alguno de los algoritmos anteriores a fin de usarlo para determinar si un grafo es o no conexo?
- Enunciar un algoritmo para determinar el número de componentes conexas de un grafo.

### Ejercicio 15

- Determinar la complejidad del algoritmo de **Prim**.
- Escribir un programa para el algoritmo de **Prim**.
- Qué técnica utiliza este algoritmo?

### Ejercicio 16

Cual es el tiempo de ejecución de BFS si su grafo de entrada se prepresenta a través de una matriz de adyacencias? Comparar con el tiempo de ejecución de en el caso de que la entrada se represente con lista de adyacencias. Vale la pena transformar la entrada?

### Ejercicio 17

- Sea  $G = (V, E)$  un grafo conexo. Dar un algoritmo de complejidad  $O(V + E)$  para obtener un camino que pase por todos los ejes de  $G$  exactamente una vez, suponiendo que esto sea posible.
- Describe como salir de un laberinto con una gran cantidad de migas y sin pasar dos veces por un mismo pasillo.

### Ejercicio 18

Dado un grafo dirigido acíclico  $G$  y dos vértices  $v$  y  $w$ , indicar la cantidad de caminos de  $v$  a  $w$  en  $G$  (no es necesario listarlos).