

# Algoritmos Elementales de Grafos

COMP-420: Análisis de Algoritmos e Introducción a las Matemáticas Discretas

Entregar:(10 de Octubre de 2018)

**Ejercicio 1.** Dibuje un grafo unidirigido conectado tal que cada vértice esté en algún ciclo no dirigido, pero que independientemente de la orientación que se dé a las aristas (es decir que se conviertan en aristas dirigidas) el grafo no esté fuertemente conectado.

**Ejercicio 2.** Determine los árboles de búsqueda primero en profundidad (DFS) para el grafo de la Figura 1 con **G** como vértice de partida y haciendo el siguiente supuesto acerca del orden dentro de las listas de adyacencia:

- Cada lista de adyacencia está en orden alfabético.

Nota: En el diagrama del árbol de búsqueda, muestre el tiempo (paso) en el que se descubrió y término cada elemento. Por ejemplo, 4/7 indica que se descubrió en el 4 y se termino en 7.

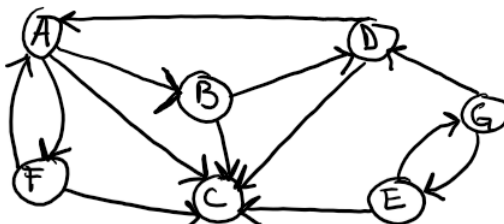


Figure 1: Grafo 1

**Ejercicio 3.** Sea  $G$  un grafo conectado, y sea  $s$  un vértice de  $G$ . Sea  $T_D$  un árbol de búsqueda DFS que se forma efectuando una búsqueda primero en profundidad en  $G$  partiendo de  $s$ . Sea  $T_B$  un árbol abarcante primero en en amplitud (BFS) que se forma efectuando búsqueda primero en amplitud en  $G$  partiendo de  $s$ . ¿Siempre se cumple que  $altura(T_D) \geq altura(T_B)$ ? ¿Importa si el grafo es dirigido o no? Presente un argumento claro o un contraejemplo.

**Ejercicio 4.** Ejecute rastreo DFS con el grafo dirigido de la Figura 2, y clasifique todas las aristas. Para esta clasificación **redibuje** el grafo que muestre los tiempos de descubrimiento y terminación de cada elemento explorado (**igual que en la nota del Ejercicio 2**). En su nuevo diagrama utilice la siguiente notación para etiquetar las aristas: (t) *tree edge*, (b) *back edge*, (c) *cross edge* y (f) *forward edge*. Para este ejercicio suponga que los vértices están indexados en orden alfabético en un arreglo *verticesAdya* y que todas las listas de adyacencia están en orden alfabético.

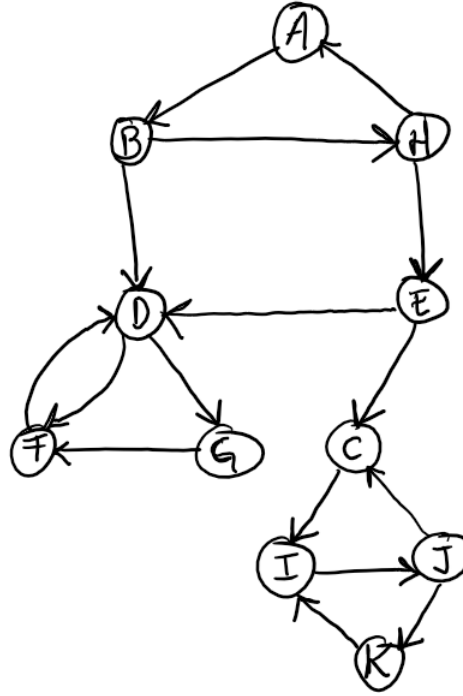


Figure 2: Grafo 2

**Ejercicio 5.** De un ejemplo de algún grafo dirigido  $G = (V, E)$ , un vértice de inicio  $s \in V$ , y un conjunto de aristas de árbol  $E_\pi \subseteq E$  tal que para cada vértice  $v \in V$ , el único camino simple en el grafo  $V, E_\pi$  de  $s$  a  $v$  es el camino más corto en  $G$ . En este ejemplo, el conjunto de aristas  $E_\pi$  debe ser imposible de generar con BFS ejecutado sobre  $G$ , sin importar el orden de los vértices en la lista de adyacencia.

**Ejercicio 6.** Si un grafo dirigido  $G$  contiene un camino de  $u$  a  $v$ , entonces  $u$  ( $u.d$ ) se descubre siempre antes de  $v$  ( $v.d$ ) en un BFS del grafo  $G$ , y por lo tanto  $v$  es un descendiente de  $u$  en el DFS forest que se produce. Lo anterior, ¿siempre se cumple? Presente un argumento claro del porque o un contraejemplo.

**Ejercicio 7.** Bosqueje las etapas clave de un algoritmo que utilizando la clasificación de aristas pueda determinar si un grafo no dirigido  $G = (V, E)$  contiene un ciclo o no. El algoritmo debe correr en tiempo  $O(V)$ , independientemente de  $|E|$ . El algoritmo debe emplear DFS.

**Ejercicio 8 (Johan).** Escribe  $P(n, k)$  la función que genera todas las particiones de  $n$  en  $k$  elementos. Por ejemplo  $P(3, 2)$  debe generar:  $\{\{0, 1\}\{2\}\}$ ,  $\{\{0\}\{1, 2\}\}$ ,  $\{\{1\}\{0, 2\}\}$