**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN**

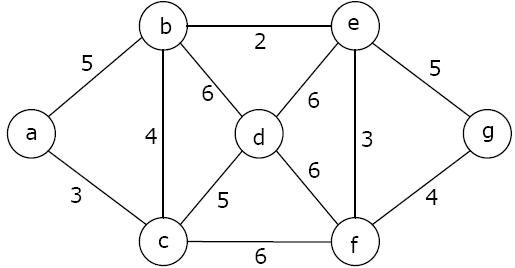
**ESTRUCTURAS DE DATOS**

**SEGUNDA EVALUACIÓN - II TÉRMINO 2018**

***Nombre: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Matrícula: \_\_\_\_\_\_\_\_\_\_\_***

**TEMA 1. (2 PUNTOS)**

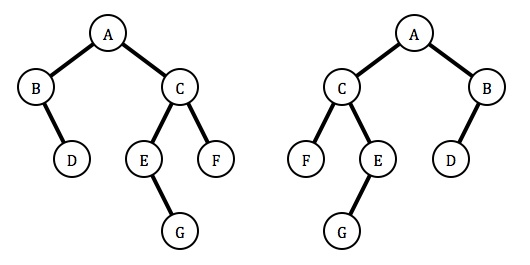
1. Dibuje el árbol AVL resultante, luego de ingresar los siguientes elementos en el orden que se presentan: [ 2 , 7 , 10 , 20 , 40 , 60 , 15, 25 , 5 , 3]
2. Cuál es el árbol de expansión mínima resultante luego de aplicar el algoritmo de Kruskal.



**TEMA 2. (10 PUNTOS)**

En la clase **AB** se le solicita implementar la función **isMirror**. La función determina si el árbol binario es espejo del árbol que se recibe como argumento.

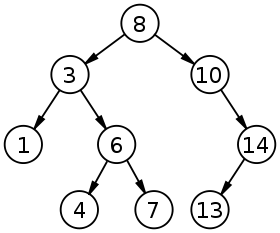
**public boolean isMirror(AB<E> arbol)**



**TEMA 3. (10 PUNTOS)**

En la clase **ABB** se le solicita implementar la función **getLevels**. La función recibe una lista de claves y retorna un mapa con las claves que se reciben de la lista y el valor es el nivel en que se encuentra cada clave.

**public Map<E,Integer> getLevels(List<E> claves)**

****

Por ejemplo:

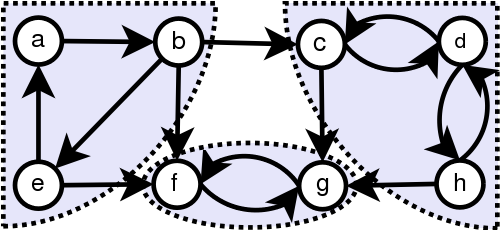
**getLevels( [13,4,10,1] )** retorna un mapa con la siguiente estructura:

**{ 13 : 3 , 4 : 3 , 10 : 1 , 1 : 2}**

**TEMA 4. (8 PUNTOS)**

En la clase **GraphLA**, usted debe implementar el método **connectedComponents** para un **Grafo Dirigido**. EL método debe retornar una lista con las componentes conexas que tiene el grafo.

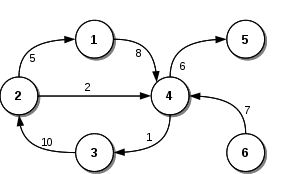
**public List<List<E>> connectedComponents()**

****

**TEMA 5. (10 PUNTOS)**

En la clase **GraphLA**, usted debe implementar el método **bestSearchFirst.** Este método explora el Grafo como una combinación de DFS y BFS, pero su criterio de selección siempre es el adyacente que tenga menor peso en su arco para continuar la exploración.

**public List<E> bestSearchFirst(E inicio)**

****

**Empezando el algoritmo desde el vértice 2.**

**[ 2 , 4 , 3 , 1 , 5 ]**

**Empezando el algoritmo desde el vértice 4.**

**[ 4 , 3 , 5 , 2 , 1 ]**

**TEMA 6. (5 PUNTOS)**

En la clase **GraphMA**, usted debe implementar el método reverse. Este método retorna un nuevo **GraphMA** con vértices nuevos que tienen la misma data del grafo original, pero el sentido de los arcos del nuevo grafo están en sentido contrario.

**public GraphMA<E> reverse()**