

EXAMEN FINAL

INF310 SX- Estructuras de Datos II. Gestión 2-2019.

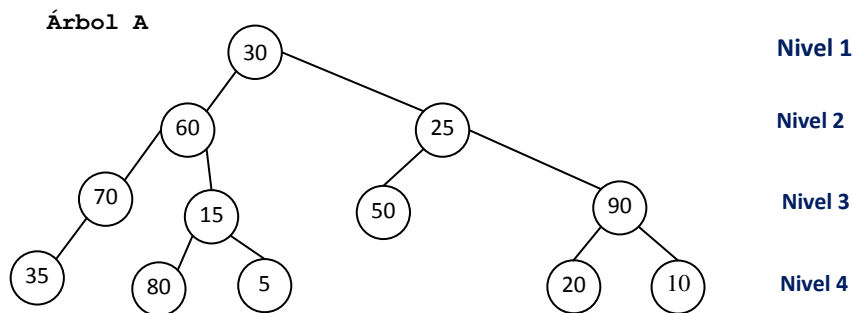
Árbol Binario

1. En la class Arbol (desordenado pero sin duplicados), escriba el procedimiento

```
public void podar(int nivel) //Asuma que nivel > 0
```

el cual pade el árbol desde el nivel dado. Si tal nivel no existe, este procedimiento no hace nada.

Por ejemplo: (En el gráfico, no se dibujan los punteros null)



A.podar (5) ;	El nivel 5 no existe: "No pasa nada" (el árbol queda igual)
A.podar (3) ; El árbol es podado desde el nivel 3 en adelante.	<p>Árbol A</p> <p>Nivel 1 Nivel 2</p>
A.podar (1) ; Podar desde el nivel 1, significa podar todo el árbol. Por tanto el árbol queda vacío (Raíz = null).	<p>Árbol A = (vacío)</p>

Grafos Dirigidos y con Peso (Taggraph)

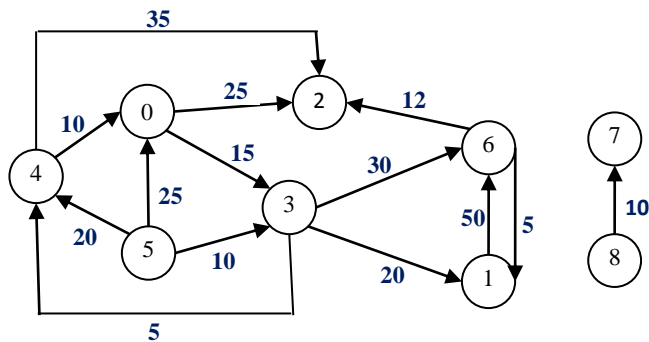
2. En la class Grafo, escriba la función

```
public int alcanza(int u, int cost[])
```

la cual partiendo de un vértice u, continúa con el vértice adyacente cuya arista tiene peso cost[0]. Si no existe esa arista, devuelve -1. Luego continua con el vértice adyacente cuya arista tiene peso cost[1]. Si no existe, devuelve -1. ...y así sucesivamente hasta completar todos los elementos del arreglo (vector) cost[].

Por comodidad, asuma que no existen dos caminos con el cost[] dado.

Por ejemplo, tomando al grafo G:



/* Para pasar un vector como parámetro, use new int[] {..}. Por ejemplo si escribimos

```
new int[] {10, 20, 50}
```

estaremos pasando el vector [10|20|50]. O sea vector[0]=10, vector[1]=20, vector[2]=50

*/

<code>alcanza(0, new int[] {15,20})=1</code>	Porque partiendo desde el 0, llegamos al 1: <div>15 20 0 ---> 3 ---> 1</div>
<code>alcanza(6, new int[] {12,25,15})= -1</code>	Partiendo desde el 6 con el 12 llegamos al 2. Pero desde el 2 no existe una arista saliente con el valor 25 (“no vale contraflecha”). Por tanto, devolvemos -1
<code>alcanza(4, new int[] {10,15,20,8})= -1</code>	Partiendo desde el 4 con el 10 llegamos al 0. Estando en el 0, con el 15 llegamos al 3. Estando en el 3, con 20 llegamos al 1. Pero, estando en el 1, no existe una arista saliente con 8. Entonces, devolvemos -1
<code>alcanza(3, new int[] {5,10,15,30})= 6</code>	<div>5 10 15 30 3 ---> 4 ---> 0 ---> 3 ---> 6</div>