**PRIMER PARCIAL**

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

**Subgrupo: A-F**

Árbol Binario

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

public void delSingle(int x)

el cual elimina a x, siempre y cuando x NO tenga 2 hijos. Si x no existe o x tiene 2 hijos, este procedimiento no hace nada.

***Recuerde que:*** *Si el nodo a eliminar es una hoja, simplemente se la elimina. Si el nodo a eliminar tiene exactamente un hijo, su único hijo ocupa su lugar.*

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

**Árbol A**

|  |  |
| --- | --- |
| **A.**delSingle**(150);** | El 150 no existe: El árbol queda igual. |
| **A.**delSingle**(**60**);** | El 60 existe, pero tiene 2 hijos: El árbol queda igual. |
| **A.**delSingle**(**80**);**  El 80 es una hoja, por lo tanto, es eliminada. | **Árbol A** |
| **A.**delSingle**(**25**);**  El 25 tiene un solo hijo (el **50**), por lo tanto, su único hijo ocupa su lugar. | **Árbol A** |

**Listas**

2. En la class Lista (no-ordenada, pero sin duplicados), escriba el procedimiento:

**public void delInto(int a, int b)**

*//Por comodidad, asuma que* **a***≠b*.

el cual elimine el número que está entre a y b. Si a o b no existen, o b aparece antes que a, o entre a y b no hay UN solo número, éste procedimiento no hace nada.

**Por ejemplo (en el main):** Por ejemplo, supongamos que la lista P = [5, 1, 2, 3, 8, 11, 10, 7, 6, 9]

P.delInto(7,**13**); // P = [5, 1, 2, 3, 8, 11, 10, 7, 6, 9] La lista P queda igual, porque el **13** no existe.

P.delInto (**20**,9); // P = [5, 1, 2, 3, 8, 11, 10, 7, 6, 9] La lista P queda igual, porque el **20** no existe.

P.delInto (**6**,**8**); // P = [5, 1, 2, 3, **8**, 11, 10, 7, **6**, 9] La lista queda igual, porque el **8** aparece antes que el **6**.

P.delInto(8,**10**); // P = [5, 1, 2, 3, 8, 10, 7, 6, 9] Entre el 8 y el 11 hay UN solo número (el 11): El 11 es eliminado.

P.delInto(3,**7**); // P = [5, 1, 2, **3**, 8, 10, **7**, 6, 9] Queda igual, porque entre el 3 y el 7 hay **más de un** número.

P.delInto(5,2); // P = [5, 2, 3, 8, 10, 7, 6, 9] Entre el 5 y el 2 hay UN solo número (el 1): El 1 es eliminado.