Ejercicios cuarta actividad del portafolio.

Asignatura: ESTRUCTURAS DE DATOS Y ALGORITMOS 15GIIN

Resolución de ejercicios teórico-prácticos sobre Recursión y Análisis de algoritmos

Calificación: 10 puntos (10% de la nota final)

Instrucciones:

- **Deberá resolver los ejercicios 1 y 2**
- Deberá enviar al profesor un archivo comprimido (.zip, .rar) con la siguiente información:
  - el nombre cumpla con la especificación siguiente: **ACT4_ PRIMERAPELLIDO_SEGUNDOAPELLIDO_NOMBRE**. Por ejemplo: ACT4_MEZA_HOUTTEMAN_OSCAR.zip
  - Los códigos de los programas en JAVA de cada ejercicio en un archivo de texto (o .java), de manera que el profesor pueda luego ejecutarlos. Coloque todas las clases e interfaces en un solo archivo por cada ejercicio.

Fecha de entrega del enunciado por parte del profesor: 20 de Diciembre 2022
Fecha de entrega de la solución por parte del alumno: 10 de Enero de 2023
Recuerde que de entregar después del 10 de Enero, se corregirá la actividad sobre 8 y no sobe 10 puntos

Se incluye el archivo con el enunciado de la actividad y el archivo BinarySearchTree.java

Preguntas:

1. Hacer un método de la clase BinarySearchTree que cuente las hojas de un árbol binario de búsqueda, usar la implementación de BinarySearchTree que se anexa a la actividad. Recuerde que una hoja es un nodo del árbol donde los apuntandores (referencias) a nodo izquierdo y derecho son nulos.
   (4 puntos)

2. Hacer un método **menoresQue** de la clase BinarySearchTree que cuente el número de elementos menores estrictos que un elemento dado, que llamamos ELEMENTO, en un árbol binario de búsqueda. ELEMENTO puede que no esté en el árbol.

   La firma del método debería ser como sigue:

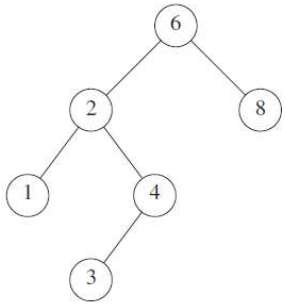
   public int menoresQue(AnyType ELEMENTO)

   Importante: El algoritmo NO tiene por qué visitar todos los nodos del árbol; y de esta forma pueda ser más eficiente. Por lo que aplicar directamente una visita en pre-orden, in-orden o post-orden e ir contando los elementos menores a ELEMENTO no es válido. Y debe utilizar la implementación dada de la clase BinarySearchTree sin aumentar la estructura (es decir, no añadir campos adicionales como, por ejemplo, agregar un campo en la clase BinaryNode que corresponda al número de nodos del árbol del cual él es raíz)

   Ayuda: El algoritmo debería ser más o menos:

   Si el árbol es vacío devolver 0.
   Si ELEMENTO es igual al elemento en la raíz del árbol entonces contar el número de nodos del árbol izquierdo y devolver ese número.
   Si ELEMENTO es menor al elemento en la raíz del árbol entonces recursivamente devolver el número de elementos menores que ELEMENTO en el árbol izquierdo.
   Si ELEMENTO es mayor que la raíz del árbol, el número de elementos menores que ELEMENTO serán 1 (por la raíz) más número de nodos del árbol izquierdo más el número que resulte de calcular recursivamente el número de nodos menores que ELEMENTO en el árbol derecho. Devolver esa suma.

   Ejemplo: el número de elementos menores que 4 en el siguiente árbol es 3 (los elementos 1, 2 y 3)



   (6 puntos)

   Recordad documentar correctamente los programas como os indiqué por email, esto valdrá 2 puntos sobre los 10 de la actividad!!