Nombre y Apellido: ...................................... N° Legajo: .............

#### Primer Parcial de Estructuras de Datos y Algoritmos

#### Segundo Cuatrimestre de 2016

|  |  |  |  |
| --- | --- | --- | --- |
| Ejercicio 1 | ***Ejercicio 2*** | ***Ejercicio 3*** | Nota |
|  |  |  |  |

Condición Mínima de Aprobación: Tener por lo menos dos ejercicios con B-

###### Consideraciones a tener en cuenta. MUY IMPORTANTE

###### El ejercicio que no respete estrictamente el enunciado será anulado.

* **puede entregar el examen escrito en lápiz**

###### Se tendrán en cuenta la eficiencia y el estilo de programación.

###### Los teléfonos celulares deben estar apagados.

## Ejercicio 1

public class BST<T> implements BinarySearchTree<T> {

private Node root;

private Comparator<? super T> cmp;

public BST(Comparator<? super T> cmp) {

this.root = null;

this.cmp = cmp;

}

public void add(T value) {

root = add(root, value);

}

private Node add(Node node, T value) {

...

}

private class Node {

T value;

Node left;

Node right;

Node(T value) {

this.value = value;

}

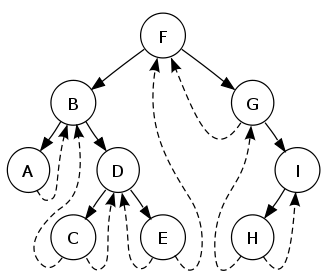
}

}

Un "Threaded Binary Tree" es un árbol binario tal que

* 1. si el hijo derecho de un nodo es null apunta al sucesor inorder del nodo
  2. si el hijo izquierdo es null, apunta al predecesor inorder del nodo
  3. las inserciones y borrados se realizan de la misma forma que en un BST, pero actualizando -de ser necesario- las referencias al sucesor y predecesor inorder.

Ejemplo



Basándose en la clase BST crear la clase ThreadedBT, que contenga los siguientes métodos

1. inOrder que no reciba parámetros e imprima el valor de los nodos en forma inorder.
2. add, que recibe un valor y lo inserta en forma ordenada. Si el valor ya estaba, no hace nada.

Ejemplo: si se insertan las claves F, A, D y C el árbol, en cada paso, sería

## Ejercicio 2

## Agregar a la clase del punto anterior un método de instancia que reciba un árbol X (del mismo tipo) y retorne true si X es subárbol de la instancia.

Ejemplo

Dado el siguiente árbol:

Los siguientes son subárboles del mismo

Los siguientes no son subárboles

## Ejercicio 3

Una Bag es una colección de datos no ordenada que almacena elementos, admitiendo elementos repetidos.

Implementar esta estructura, respetando la siguiente interfaz. La clase debe recibir un comparador por constructor.

**public interface Bag<T> {**

/\*\* Agrega el elemento

\*/

**public void add(T value);**

/\*\* Devuelve la cantidad de veces que el elemento pasado por parámetro existe en la colección. \*/

**public int occurrencesOf(T value);**

/\*\* Elimina el elemento que más veces aparece en ese momento y lo devuelv­­­e. \*/

**public T removeMostPopular();**

**}**

No se puede utilizar la API de Java en este ejercicio.

**El método add y occurrencesOf deben ser resueltos en O(n) mientras que removeMostPopular debe ser en O(1).**