

TRABAJO PRÁCTICO N ° 7

Árboles binarios

Departamento de Ciencias e Ingeniería de la Computación - U.N.S.

Ejercicio 1:

- Programa el *TDA ÁrbolBinario* respetando la interfaz *BinaryTree<E>* provista por la cátedra. El árbol deberá contar con una estructura con enlaces a cada uno de sus hijos y al padre, donde los rótulos de los nodos son de tipo genérico E.
- Corra el correspondiente test JUnit para verificar la correctitud de su implementación.
- Calcule el orden del tiempo de ejecución de cada una de las operaciones de la clase árbol binario. Justifique adecuadamente.

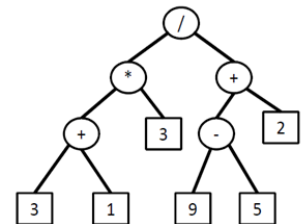
Ejercicio 2:

Agregue un método a la clase árbol binario programada en el inciso anterior tal que recorra el árbol en pre-orden y retorne un diccionario donde sus entradas tengan como clave al rótulo del padre y como valor a los rótulos de cada uno de sus hijos. Los rótulos ubicados en hojas del árbol no deben pertenecer al diccionario, de esta forma en el diccionario no pueden existir valores nulos.

Ejercicio 3:

Un árbol binario de expresión es un árbol binario donde cada uno de sus nodos internos están rotulados con operadores y sus hojas con operandos (tal como el que se muestra en la figura). 1

Dado un árbol binario A que representa una expresión aritmética, escriba un método recursivo (es decir, no puede usar los iteradores) que retorne un iterable de caracteres con la notación infija de la expresión que el árbol representa.



Ejercicio 4:

- Escriba un método cuya signature sea: `public void completarDerechos(E r, BinaryTree<E> t) throws EmptyTreeException` tal que agregue como hijos derechos (de aquellos nodos que solo tengan hijo izquierdo) nodos con rótulos *r*. Este método deberá lanzar `EmptyTreeException` en caso de que el árbol *t* se encuentre vacío.
- Indique el orden del tiempo de ejecución de su solución. Justifique adecuadamente.

Ejercicio 5:

Agregue un método a la clase definida en el ejercicio 1 cuya signature sea: `public void eliminarSubarbol(Position<E> p) throws InvalidPositionException`. Este método deberá eliminar del árbol receptor del mensaje todo el subárbol con raíz *p*. Recuerde que para resolver este ejercicio cuenta con acceso total a la estructura.