



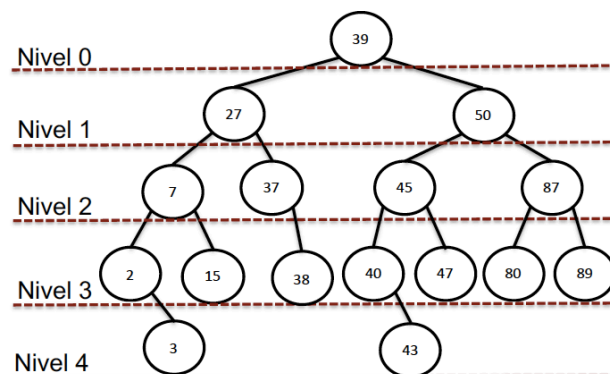
UNIVERSIDAD DEL CAUCA -
FACULTAD DE INGENIERIA ELECTRONICA Y TELECOMUNICACIONES
PROGRAMA DE INGENIERIA DE SISTEMAS
CURSO DE LABORATORIO DE ESTRUCTURAS DE DATOS II: GRUPO A

Práctica 7: Árboles AVL

En esta práctica realizaremos la implementación en java de un árbol AVL, con el fin de afianzar los conocimientos adquiridos y generar destrezas sobre la utilización de este tipo de estructura. Para la realización de la misma trataremos de utilizar lo visto en clase de teoría.

Recuerde que:

1. Los árboles AVL, son árboles ordenados o de búsqueda que además cumplen las condiciones de balanceo para cada uno de sus nodos.
2. Es una de las primeras estructuras de datos de árboles de búsqueda que son equilibradas y reciben el nombre de AVL por Adelson-Velskii-Landis.
3. Un árbol AVL es un árbol binario de búsqueda en el que las alturas de los subárboles izquierdos y derecho de cualquier nodo difieren a lo sumo en 1. Esta restricción impuesta sobre la altura de los subárboles de un árbol AVL se le conoce como propiedad de los árboles AVL, y debe ser cumplida por todos y cada uno de los nodos del árbol.



Funciones Para Implementar

1. Cree un árbol AVL a partir de un archivo e imprimirlo.
2. Recorridos del árbol (Inorden, Postorden, Preorden).
3. Retornar cual es el peso del árbol. Recuerde que el peso de un árbol es el número de hojas.
4. Cree una funcionalidad que permita determinar si dos elementos del árbol son primos hermanos. Un elemento A es primo hermano de B, si se encuentran en el mismo nivel de árbol y si B es hijo del padre de A.

Elementos	Información
27 - 50	No son primos, son Hermanos.
7 - 45	Son primos hermanos.
2 - 40	No son primos hermanos.
3 - 43	No son primos hermanos.

5. Cree una funcionalidad que permita mostrar todos los primos hermanos de un elemento ingresado.

Elemento	Información
27	No tiene primos hermanos.
7	Los primos hermanos son: 45, 87
2	Los primos hermanos son: 38
3	No tiene primos hermanos.



UNIVERSIDAD DEL CAUCA -
FACULTAD DE INGENIERIA ELECTRONICA Y TELECOMUNICACIONES
PROGRAMA DE INGENIERIA DE SISTEMAS
CURSO DE LABORATORIO DE ESTRUCTURAS DE DATOS II: GRUPO A

Implementación Básica

```
//Clase Nodo
public class Nodo <T> {
    private Nodo izq;    // hijo izquierdo
    private Nodo der;    // hijo derecho
    private int  factorEquilibrio;    // factor de equilibrio
    private T elemento;    // los datos  como elementos del arbol avl
// Constructores
    public Nodo(T elemento) {
        this(elemento, null, null );
    }
    public Nodo( T elemento, Nodo izq, Nodo der )
    {
        this.elemento = elemento;
        this.izq = izq;
        this.der = der;
        factorEquilibrio = 0;
    }
//Métodos Getters and Setters
    -----
    -----
}

//Clase Árbol
public class Arbol {
    private Nodo raiz;
    public Arbol() {
        this.raiz = null;
    }
    public Nodo getRaiz() {
        return raiz;
    }
    public void insertar(Comparable elemento ) {
        raiz = insertar(raiz,elemento );
    }
    public static int factorEquilibrio(Nodo arbol){
        if (arbol==null)
            return 0;
        else
            return (arbol.getFactorEquilibrio()+1);
    }
    private static int mFE( int alturaIzquierdo, int alturaDerecho ) {
        if (alturaIzquierdo > alturaDerecho)
            return alturaIzquierdo;
        else
            return alturaDerecho;
    }
}
```



UNIVERSIDAD DEL CAUCA -
FACULTAD DE INGENIERIA ELECTRONICA Y TELECOMUNICACIONES
PROGRAMA DE INGENIERIA DE SISTEMAS
CURSO DE LABORATORIO DE ESTRUCTURAS DE DATOS II: GRUPO A

```
private Nodo insertar(Nodo arbol,Comparable elemento){
    if( arbol == null ){
        arbol = new Nodo(elemento, null, null);
    }
    else if (elemento.compareTo(arbol.getElemento()) < 0 ) {
        arbol.setIzq(insertar(arbol.getIzq(), elemento));
        if (factorEquilibrio(arbol.getDer()) - factorEquilibrio(arbol.getIzq()) == -2 ){
            if (elemento.compareTo(arbol.getIzq().getElemento()) < 0 )
                arbol = RotacionSimpleDer(arbol);
            else
                arbol = RotacionDobleIzq_Der(arbol);
        }
    }
    else if (elemento.compareTo(arbol.getElemento()) > 0 ){
        arbol.setDer(insertar(arbol.getDer(), elemento));
        if( factorEquilibrio(arbol.getDer()) - factorEquilibrio(arbol.getIzq()) == 2){
            if( elemento.compareTo(arbol.getDer().getElemento()) > 0 )
                arbol = RotacionSimpleIzq(arbol);
            else
                arbol = RotacionDobleDer_Izq(arbol);
        }
    }
    else{
        /// En el caso de que sean valores duplicados no hace nada
    }
    arbol.setFactorEquilibrio(mFE(factorEquilibrio(arbol.getIzq()),
factorEquilibrio(arbol.getDer())));
    return arbol;
}

private static Nodo RotacionSimpleDer(Nodo arbol) {
    Nodo subArbol = arbol.getIzq();
    arbol.setIzq(subArbol.getDer());
    subArbol.setDer(arbol);
    arbol.setFactorEquilibrio(mFE(factorEquilibrio(arbol.getIzq()),
factorEquilibrio(arbol.getDer())));
    subArbol.setFactorEquilibrio(mFE(factorEquilibrio(subArbol.getIzq()),
arbol.getFactorEquilibrio()));
    return subArbol;
}

private static Nodo RotacionSimpleIzq(Nodo arbol){
    Nodo subArbol = arbol.getDer();
    arbol.setDer(subArbol.getIzq());
    subArbol.setIzq(arbol);
    arbol.setFactorEquilibrio(mFE(factorEquilibrio(arbol.getIzq()),
factorEquilibrio(arbol.getDer())));
    subArbol.setFactorEquilibrio(mFE(factorEquilibrio(subArbol.getDer()),
arbol.getFactorEquilibrio()));
    return subArbol;
}
```



UNIVERSIDAD DEL CAUCA -
FACULTAD DE INGENIERIA ELECTRONICA Y TELECOMUNICACIONES
PROGRAMA DE INGENIERIA DE SISTEMAS
CURSO DE LABORATORIO DE ESTRUCTURAS DE DATOS II: GRUPO A

```
private static Nodo RotacionDobleIzq_Der(Nodo arbol){  
    arbol.setIzq(RotacionSimpleIzq(arbol.getIzq()));  
    return RotacionSimpleDer( arbol );  
}  
private static Nodo RotacionDobleDer_Izq(Nodo subArbol){  
    subArbol.setDer(RotacionSimpleDer(subArbol.getDer()));  
    return RotacionSimpleIzq(subArbol);  
}  
}
```