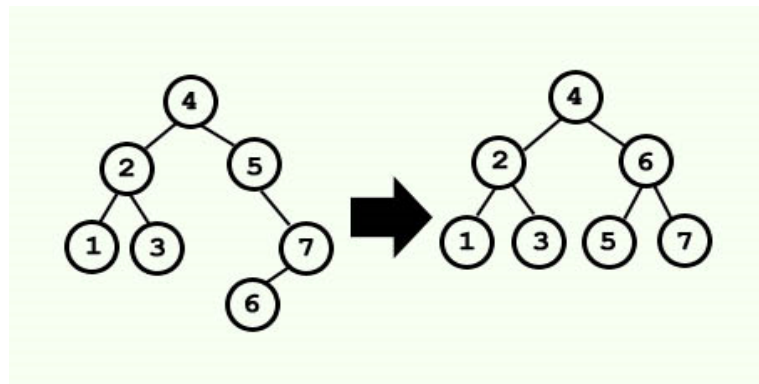


Documentacion - desarrollador arbol AVL

Irarrázabal Callejas Norton John
Norton.Dante.I@gmail.com
Facultad de Ciencias
Universidad de La Serena



Índice

1. Análisis del arbol AVL	3
2. Ejemplos	4
3. Implementación arbol AVL	5

1. Análisis del arbol AVL

Definición. Un árbol AVL es un árbol binario de búsqueda que cumple con la condición de que la diferencia entre las alturas de los subárboles de cada uno de sus nodos es, como mucho 1, es decir balanceado.

Recordamos que un árbol binario de búsqueda consiste en que cada nodo cumple con que todos los nodos de su subárbol izquierdo son menores que la raíz y todos los nodos del subárbol derecho son mayores que la raíz.

El tiempo de las operaciones sobre un árbol binario de búsqueda son $O(\log n)$ promedio, pero el peor caso es $O(n)$, donde n es el número de elementos.

La propiedad de equilibrio que debe cumplir un árbol para ser AVL asegura que la profundidad del árbol sea $O(\log(n))$, por lo que las operaciones sobre estas estructuras no deberán recorrer mucho para hallar el elemento deseado.

El tiempo de ejecución de las operaciones sobre estos árboles es, a lo sumo $O(\log(n))$ en el peor caso, donde n es la cantidad de elementos del árbol. Sin embargo, y como era de esperarse, esta misma propiedad de equilibrio de los árboles AVL implica una dificultad a la hora de insertar o eliminar elementos. Ya que es probable que estas operaciones afecten la propiedad de equilibrio.

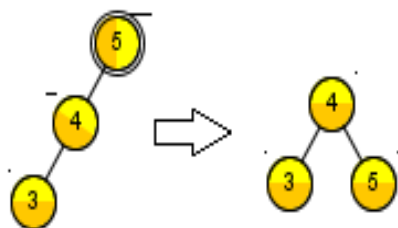
Por lo que se efectúan operaciones sobre el árbol conocidas como rotaciones, que nos ayudan a conservar el orden y a restaurar la propiedad de equilibrio de el árbol AVL.

Factor de equilibrio o propiedad de equilibrio : $FE = \text{altura subarbol derecho} - \text{altura sub arbol izquierdo}$.

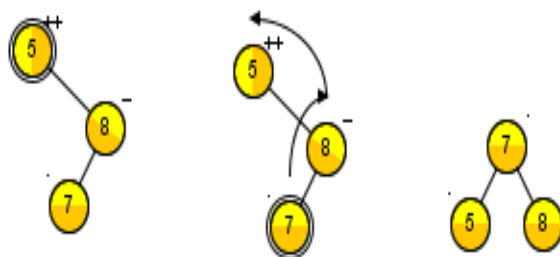
Se usan 4 tipos de rotaciones.

- Rotacion Simple a la derecha: Se usa cuando el subarbol izquierdo de un nodo sea 2 unidades mas alto que el derecho, es decir que este cargado a la izquierda. $FE (= -2)$
- Rotacion Simple a la izquierda: Se usa cuando subarbol derecho de un nodo sea 2 unidades mas alto que el izquierdo, es decir que este cargado a la derecha. ($FE=2$)
- Rotacion Doble a la derecha: Si $FE > 1$ y su hijo derecho tiene signo -. Esto es el equivalente a i: aplicar rotacion simple a la derecha. ii: aplicar rotacion simple a la izquierda.
- Rotacion Doble a la izquierda: Si $FE < -1$ y su hijo izquierdo tiene signo +. i: aplicar rotacion simple a la izquierda. ii: aplicar rotacion simple a la derecha.

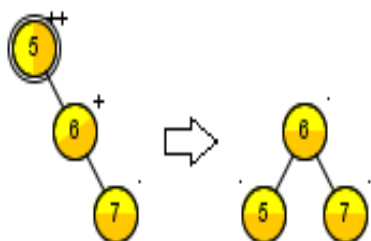
2. Ejemplos



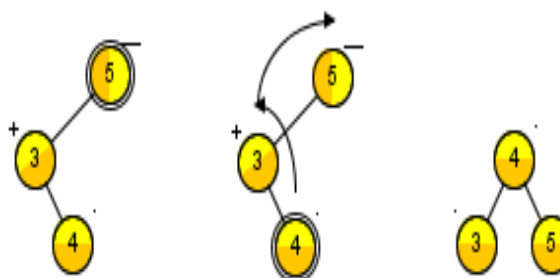
Rotación simple a la derecha.



Rotacion doble a la derecha.



Rotacion simple a la izquierda.



Rotacion doble a la izquierda.

3. Implementación arbol AVL

El arbol avl se elaboro en Eclipse Jee Oxygen es una plataforma de software compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrolladores java.

Pueden existir diversas formas de implementar un arbol AVL. Sin embargo, la implementación que yo hice para los nodos del arbol AVL es la siguiente:

```
public class Nodo_arbol_AVL {
    int Valor , FE;
    Nodo_arbol_AVL hijoizquierdo , hijoderecho;

    public Nodo_arbol_AVL (int Valor1) {
        this.Valor=Valor1;
        this.FE=0;
        this.hijoizquierdo=null;
        this.hijoderecho=null;
    }
}
```

Los nodos pueden tener muchos mas datos ya sean String, int, double etc. Sin embargo por simplicidad solo usaremos un dato de tipo entero, ahora si lo queremos con mas datos bastaria con incorporar otra variable.

Cada nodo tendra su propio factor de equilibrio, para poder evaluar el balance del arbol AVL. Ademas de un hijo izquierdo y un hijo derecho de su misma clase.

Ahora el arbol AVL lo defini como APLAVL y este tendra solamente el nodo raiz.

```
public class APLAVL{
    private Nodo_arbol_AVL raiz;

    public APLAVL() {
        raiz=null;
    }
}
```

Ahora recordemos que cada una de estas clases tiene múltiples metodos sin embargo no es menester nombrarlos todos en este documento, para esta causa incorpore diagramas UML y el Javadoc en donde podran encontrar todos los metodos.

Referencias

- [1] Árboles AVL: rua.ua.es/dspace/handle/10045/16037