ITAM

Estructura de Datos Avanzadas

Prof.: Fernando Esponda

Tarea: Alturas Promedio

Guillermo Arredondo Renero

C.U.: 000197256

**Introducción. Planteamiento del problema a resolver.**

Con el objetivo de seguir estudiando las implementaciones de los árboles binarios de búsqueda y analizar las diferencias de eficiencia entre los árboles generales y los árboles balanceados, se desea comparar el promedio de las alturas que se generan en los árboles binarios de búsqueda al insertar *n* cantidad de datos. Es importante recordar que la característica más importante de los árboles balanceados AVL es su capacidad de compactar la estructura de datos completa, lo que permite un menor recorrido de nodos enlazados y, consecuentemente, una reducción en la complejidad de operaciones. Dado a que los árboles binarios AVL se reestructuran constantemente con el objetivo de mantener una diferencia menor a dos entre las alturas de los subárboles izquierdo y derecho, estos árboles reducen su altura; por lo tanto, de estar bien implementados, en el experimento siguiente podrá observarse claramente una diferencia significante entre las alturas de los árboles generales y los árboles balanceados.

**Investigación previa.**

Más allá de la certeza de que un árbol AVL, por su propia naturaleza, tenga una altura mucho menor que la de los árboles binarios de búsqueda comunes, y considerando que su complejidad de operaciones está regida por O(lg(n)) para cualquiera de los casos, resulta valido estimar que la diferencia entre alturas será representada por una constante aproximada a la base del logaritmo, es decir, dos. Esto es fácil verificarlo, pues en el caso de los árboles balanceados se asegura una distribución del *n/*2 por cada subrama. De hecho, investigando sobre el desarrollo e implementación de los árboles balanceados AVL, la doctora Guardati expresa en su libro, *Estructuras de Datos,* que múltiples estudios han demostrado que la altura de los árboles AVL mantienen una relación con la serie de Fibonacci y con el número de elementos contenidos. Además, se considera que un árbol AVL con *n* elementos jamás sobrepasara la altura de 1.44*log(n). A lo largo del experimento comprobaremos este hecho.

**Diseño de experimento.**

Una vez implementadas las clases *BinarySearchTree<T>* y *BinarySearchTreeAVL<T>* con sus respectivos métodos y pruebas independientes para asegurar el funcionamiento correcto de cada una de estas estructuras, se diseñó un experimento a través del cual se utilizaran dos ciclos *for* anidados para la construcción de árboles y el cálculo y registro de sus respectivas alturas. Con el objetivo de presentar los datos de la forma más detallada posible se creó una función auxiliar que construyera árboles aleatorios a través de la librería *Random,* la cual recibía la interface de árboles implementada y el número de elementos que se deseaban insertar. Por consecuente, fue necesario realizar 30 pruebas de creaciones de árboles para poder obtener resultados aproximados con la menor variación de error posible.

De este modo, los ciclos anidados resolvían los problemas de la cantidad de datos, el promedio de las 30 creaciones de árboles diferentes y, a su vez, se pensó importante mantener un registro de las alturas de cada árbol creado para fines estadísticos. El experimentó fue el siguiente:

```java
public static void creaArbolRandom(BinarySearchTreeADT<Double> arbol, int n){
    Double elem;
    Random rand = new Random();
    for(int i = 1; i<=n; i++){
        elem = rand.nextInt(100000) + rand.nextDouble();
        arbol.add(elem);
    }
}
public static void main(String[] args) {
    //PRUEBAS
    BinarySearchTreeADT<Double> arbol = new BinarySearchTree();
    BinarySearchTreeADT<Double> arbolAVL = new BinarySearchTreeAVL();
      for(int i = 1; i<10; i++){
```

```java
long sumaAr, sumaArBal, prom1, prom2;
int altura1, altura2;
ArrayList<Integer> listaAr = new ArrayList();
ArrayList<Integer> listaArBal = new ArrayList();
for(int n = 1; n<10000000; n*=5){
    sumaAr = 0;
    sumaArBal = 0;

    for(int j = 1; j<=30; j++){
        arbol = new BinarySearchTree();
        arbolAVL = new BinarySearchTreeAVL();
        creaArbolRandom(arbol, n);
        creaArbolRandom(arbolAVL, n);
        altura1=arbol.altura();
        altura2=arbolAVL.altura();
        listaAr.add(altura1);
        listaArBal.add(altura2);
        sumaAr+=altura1;
        sumaArBal+=altura2;
    }
    prom1 = sumaAr/30;
    prom2 = sumaArBal/30;
    System.out.println("\nPromedio de Alturas para arboles de "+n+" elementos:");
    System.out.println("\tNormal: "+prom1+"\nDatos: "+listaAr.toString());
    System.out.println("\tBalanceado: "+prom2+"\nDatos: "+listaArBal.toString());
    listaAr.clear();
    listaArBal.clear();
}
```

## Resultados y análisis de resultados.

```
Promedio de Alturas para arboles de 1 elementos:
	Normal: 1
Datos: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
	Balanceado: 1
Datos: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

Promedio de Alturas para arboles de 5 elementos:
	Normal: 3
Datos: [4, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 4, 4, 5, 3, 3, 4, 4, 3, 4, 3, 3, 3, 4, 4, 3, 4, 4, 3]
	Balanceado: 3
Datos: [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3]

Promedio de Alturas para arboles de 25 elementos:
	Normal: 8
Datos: [8, 8, 11, 9, 10, 9, 11, 9, 7, 7, 10, 8, 8, 6, 9, 10, 7, 7, 9, 8, 11, 10, 9, 8, 7, 9, 7, 8, 8, 8]
	Balanceado: 5
Datos: [6, 7, 6, 5, 5, 5, 5, 6, 6, 6, 7, 5, 6, 6, 5, 6, 5, 7, 5, 5, 6, 5, 6, 5, 5, 5, 5, 5, 5, 5]

Promedio de Alturas para arboles de 125 elementos:
	Normal: 14
Datos: [13, 13, 15, 13, 16, 14, 13, 16, 15, 14, 15, 16, 12, 14, 15, 13, 12, 11, 14, 14, 14, 13, 13, 16, 18, 13, 13, 14, 15, 13]
	Balanceado: 8
Datos: [9, 7, 8, 10, 9, 10, 9, 8, 8, 9, 8, 7, 7, 9, 9, 8, 8, 7, 6, 9, 8, 8, 7, 7, 8, 8, 10, 9, 7, 8]

Promedio de Alturas para arboles de 625 elementos:
	Normal: 20
Datos: [19, 21, 20, 23, 22, 19, 22, 21, 23, 22, 20, 21, 20, 18, 18, 18, 20, 23, 22, 18, 23, 17, 18, 23, 22, 20, 21, 20, 22, 19]
	Balanceado: 10
Datos: [8, 8, 14, 10, 10, 9, 9, 10, 11, 9, 9, 11, 10, 11, 11, 11, 10, 12, 12, 12, 11, 11, 11, 13, 9, 10, 10, 13, 10, 8]

Promedio de Alturas para arboles de 3125 elementos:
	Normal: 26
Datos: [25, 25, 27, 27, 27, 29, 26, 26, 27, 24, 26, 26, 25, 28, 27, 24, 27, 24, 28, 25, 25, 25, 27, 27, 27, 24, 25, 27, 27, 23]
	Balanceado: 13
Datos: [14, 11, 8, 14, 14, 15, 13, 15, 10, 13, 12, 16, 16, 13, 18, 10, 11, 14, 12, 14, 17, 13, 15, 12, 12, 11, 14, 11, 14, 14]
```

```
Promedio de Alturas para arboles de 3125 elementos:
        Normal: 26
Datos: [25, 25, 27, 27, 27, 29, 26, 26, 27, 24, 26, 26, 25, 28, 27, 24, 27, 24, 28, 25, 25, 25, 27, 27, 27, 24, 25, 27, 27, 23]
        Balanceado: 13
Datos: [14, 11, 8, 14, 14, 15, 13, 15, 10, 13, 12, 16, 16, 13, 18, 10, 11, 14, 12, 14, 17, 13, 15, 12, 12, 11, 14, 11, 14, 14]

Promedio de Alturas para arboles de 15625 elementos:
        Normal: 33
Datos: [34, 35, 33, 33, 31, 35, 32, 37, 32, 35, 35, 37, 33, 32, 30, 31, 33, 34, 33, 39, 32, 31, 35, 33, 34, 32, 37, 32, 35, 34]
        Balanceado: 15
Datos: [19, 13, 15, 14, 9, 15, 15, 17, 15, 15, 19, 14, 17, 15, 17, 16, 14, 11, 13, 12, 13, 18, 16, 17, 17, 18, 13, 16, 19, 15]

Promedio de Alturas para arboles de 78125 elementos:
        Normal: 39
Datos: [37, 38, 41, 40, 43, 38, 41, 37, 37, 38, 44, 43, 37, 39, 40, 40, 38, 40, 37, 37, 42, 38, 41, 38, 41, 40, 38, 42, 39, 38]
        Balanceado: 17
Datos: [20, 15, 17, 16, 17, 19, 16, 18, 19, 15, 18, 18, 19, 15, 16, 17, 20, 16, 15, 18, 14, 20, 21, 19, 21, 19, 18, 14, 18, 18]

Promedio de Alturas para arboles de 390625 elementos:
        Normal: 46
Datos: [43, 47, 46, 47, 46, 48, 49, 43, 48, 48, 48, 46, 46, 48, 46, 47, 48, 50, 43, 47, 49, 50, 44, 45, 45, 46, 46, 46, 48, 47]
        Balanceado: 19
Datos: [18, 15, 20, 22, 22, 22, 22, 15, 16, 16, 18, 16, 17, 26, 25, 25, 21, 18, 21, 16, 16, 22, 24, 20, 19, 21, 18, 18, 20, 23]

Promedio de Alturas para arboles de 1953125 elementos:
        Normal: 53
Datos: [51, 53, 50, 55, 53, 53, 51, 56, 56, 50, 52, 53, 53, 52, 57, 57, 55, 53, 55, 53, 51, 53, 52, 52, 52, 49, 54, 54, 54, 55]
        Balanceado: 22
Datos: [19, 26, 23, 22, 23, 25, 19, 23, 22, 17, 21, 22, 27, 23, 25, 19, 23, 18, 19, 24, 26, 17, 26, 19, 21, 37, 17, 18, 20, 29]
BUILD STOPPED (total time: 6 minutes 24 seconds)
```

Como puede observarse, las alturas de los árboles balanceados fueron siempre menores o iguales que aquellas de los árboles generales. Conforme aumentó el número de datos, la diferencia entre los promedios de altura se fue ampliando, mientras que, en volúmenes de datos muy chicos, como son *n*=1 o *n*=5 los promedios incluso fueron iguales. No obstante, al comparar esta diferencia con respecto a cantidad como *n*=1953125, observamos que los promedios de altura difieren notablemente, la altura media de los árboles binarios de búsqueda supera por más del doble a aquellas de los árboles AVL.

Asimismo, puede observarse que la distribución entre las alturas con respecto a la media adquiere mayor similitud, aproximación y repetición en los casos de árboles generales; por ejemplo, tomemos *n*=3125 elementos:

Normal: 26

Datos: [25, 25, 27, 27, 27, 29, 26, 26, 27, 24, 26, 26, 25, 28, 27, 24, 27, 24, 28, 25, 25, 25, 27, 27, 27, 24, 25, 27, 27, 23]

Moda: altura 27 (10 veces repetida) Varianza: 2

Menor: altura 23 Mayor: altura 29 Datos menores a la media: 12

Balanceado: 13

Datos: [14, 11, 8, 14, 14, 15, 13, 15, 10, 13, 12, 16, 16, 13, 18, 10, 11, 14, 12, 14, 17, 13, 15, 12, 12, 11, 14, 11, 14, 14]

Moda: altura 14 (8 veces repetida) Varianza: 4.69

Menor: altura 8 Mayor: altura 18 Datos menores a la media: 10

En cierto modo, la mayor distribución en las alturas registradas en los árboles AVL indica que sus organizaciones son más diversas; sin embargo, el sentido común indicaría lo contrario. Los árboles balanceados están diseñados de tal forma de que sus nodos se encuentren lo más aglomerados posible, por lo que ante el mismo número de datos de entrada deberían obtenerse alturas más similares, pues todas tienden a compactarse lo más posible. En cambio, los árboles binarios de búsqueda general, pueden sufrir una gran variedad de casos, desde aquellos en que se encuentren balanceados hasta aquellos en los que el árbol resulte una lista enlazada, por lo que su estadística debería mostrar una varianza mucho mayor que la de los árboles balanceados. Es muy probable que este error estadístico se deba a los errores de generación de número aleatorios.

Por último, cabe señalar que para cada *n* cantidad de datos, la hipótesis del libro de la Doctora Guardati se cumplió, la altura de los árboles AVL no superan la cantidad de 1.44*log(n). De hecho, en su gran mayoría de los casos, el promedio de altura se encuentra ligeramente por encima del valor del log(n).