

Arboles Binarios de Búsqueda Balanceados

Práctica 14 Arboles Balanceados

Objetivo

Esta práctica tiene como objetivo que el alumno refuerce sus conocimientos sobre el tema de árboles balanceados, en particular sobre los conocidos como árboles AVL. El alumno implementará una de las operaciones de dicha estructura de datos.

Descripción general

Un árbol AVL es un árbol binario de búsqueda balanceado. Un árbol está balanceado si para todo nodo en el árbol, la diferencia entre las alturas de sus dos subárboles es a lo más de una unidad. Así que un árbol AVL debe mantener una relación de orden entre sus nodos, pues es un árbol binario de búsqueda, y además debe estar balanceado.

La práctica consiste en completar la clase `ArbolAvl` con la implementación de la operación para eliminar elementos. La implementación de esta función debe ser de manera recursiva por lo que se sugiere se use un método privado.

Se debe asegurar que, después de eliminar un elemento, el árbol siga siendo un árbol AVL. Esto quiere decir que la operación de eliminación propia de un árbol binario de búsqueda ahora deben incluir la funcionalidad de rebalanceo, lo que puede requerir operaciones de rotaciones simples o dobles.

Material

El material de esta práctica consta de los siguientes archivos:

- **NodoAvl.class** clase que implementa los nodos para árboles AVL.
- **ArbolAvl.java** clase que implementa la estructura de datos árboles AVL. Aquí se implementará las operación para eliminar elementos del árbol.
- **PruebaAvl.class** programa para probar la clase `ArbolAvl`.
- **PruebaAvl\$ComparaEnteros.class** clase interna de la clase `PruebaAvl` que implementa un comparador de objetos enteros. Servirá para hacer pruebas con árboles AVL que trabajen con objetos enteros.

- Documentación:
 - `NodoAvl.html` documentación de la clase `NodoAvl`.
 - `ArbolAvl.html` documentación de la clase `ArbolAvl` que se proporciona.

Desarrollo

1. Descargar los archivos `NodoAvl.class`, `ArbolAvl.java`, `PruebaAvl.class` y `PruebaAvl$ComparaEnteros.class` en el directorio donde se va a desarrollar la práctica.
2. Leer la documentación y revisar el material que se proporciona, especialmente el archivo `ArbolAvl` que es la clase que implementa árboles balanceados. Recordar que tanto la firma de los métodos que se proporcionan como la de los métodos que se solicitan en el desarrollo de la práctica, no podrán ser modificados.
3. Completar la clase `ArbolAvl` con el siguiente método:
 - a. **eliminar(Object elem)** Método encargado de eliminar el elemento especificado como parámetro y mantener el árbol balanceado. Llamará a un método interno recursivo que eliminará el elemento indicado siguiendo el mismo algoritmo que se usa para árboles binarios de búsqueda. Al eliminar el elemento verifica que el árbol esté balanceado y en caso necesario realiza rotaciones para mantener el equilibrio del árbol. En caso de no encontrar el elemento a eliminar, no hace nada.

Ayuda:

Pasos para eliminación de elementos en árboles AVL:

- Aplicar el algoritmo de eliminación para árboles binarios de búsqueda.
 - A partir de la hoja que se elimine se hace un recorrido hacia “arriba del árbol” para validar cada ancestro de la hoja eliminada hasta llegar a la raíz del árbol. En cada validación se actualiza la altura del nodo ancestro, se valida si está balanceado y en caso requerido se realizan las rotaciones necesarias.
4. Probar la implementación realizada ejecutando el programa `PruebaAvl`. Si el programa funciona adecuadamente se verán los siguientes mensajes:

```
-- Creando arbol vacio
Imprimiendo arbol...
El arbol esta vacio.

Eliminando elemento inexistente...
Tamano del arbol: 0

-- Creando arbol con un solo elemento: { 13 }
Imprimiendo arbol...
13:0

Tamano del arbol: 1

Eliminando elemento inexistente...
Imprimiendo arbol...
13:0

Tamano del arbol: 1

Eliminando elemento existente 13...
Imprimiendo arbol...
El arbol esta vacio.

Tamano del arbol: 0

-- Creando arbol con los siguientes elementos: { 10 15 7 13 14 8 4
20 19 12 11 28 1 }
Imprimiendo arbol...
1:0
4:1
7:2
8:0
10:3
11:0
12:1
13:0
14:4
15:0
19:2
20:1
28:0

Tamano del arbol: 13

Eliminando elemento 15 (prueba caso: nodo a eliminar sin hijos)...
```

Imprimiendo arbol...

1:0
4:1
7:2
8:0
10:3
11:0
12:1
13:0
14:2
19:0
20:1
28:0

Tamano del arbol: 12

Eliminando elemento 4 (prueba caso: nodo a eliminar con un solo hijo)...

Imprimiendo arbol...

1:0
7:1
8:0
10:3
11:0
12:1
13:0
14:2
19:0
20:1
28:0

Tamano del arbol: 11

Eliminando elemento 7 (prueba caso: nodo a eliminar con dos hijos)...

Imprimiendo arbol...

1:0
8:1
10:3
11:0
12:1
13:0
14:2
19:0
20:1
28:0

Tamano del arbol: 10