

Taller de Árboles AVL: Implementación de un árbol AVL (corregido) con inserción, eliminación y visualización.

Andrés Sebastián Pinzón Gutiérrez 2221887

<https://github.com/Abysyu29/Taller-arboles-AVL.git>

Informe sobre la Implementación y Verificación del Árbol AVL

En este trabajo se implementó un árbol AVL en Python, con el objetivo de garantizar que la estructura se mantuviera balanceada después de realizar operaciones de inserción y eliminación. Se incluyeron también pruebas automáticas para verificar su correcto funcionamiento.

Implementación

Cada nodo del árbol se definió mediante la estructura `Node(value, left, right, height)`, donde la altura de una hoja se considera 1.

Se desarrollaron funciones auxiliares como `getHeight(node)` y `updateHeight(node)` para mantener actualizadas las alturas, además de `getBalance(node)` para calcular el factor de balance.

Cuando un nodo se desbalancea, el programa aplica una de las rotaciones clásicas del árbol AVL (izquierda, derecha o combinadas) para restaurar el equilibrio.

La inserción se realiza siguiendo la lógica de un árbol binario de búsqueda (BST), recalculando luego alturas y balance para aplicar las rotaciones necesarias.

Errores Detectados y Solución

Durante las pruebas iniciales se identificó un error principal: las rotaciones se ejecutaban, pero no se reasignaba la nueva raíz del subárbol.

Por ejemplo, se llamaba `rotate_right(node)` en lugar de `node = rotate_right(node)`, lo que impedía que el árbol se reorganizara correctamente.

Además, el archivo original (`Lab_AVL-error.py`) contenía un guion en su nombre, lo que impedía importarlo como módulo en Python.

Soluciones aplicadas:

Reasignación del resultado de las rotaciones para que los subárboles se actualicen correctamente.

Cambio de nombre del archivo a Lab_AVL_error.py.

Implementación completa de la eliminación (delete) con rebalanceo posterior.

Adición de funciones de inspección y visualización (inorder, print_vertical, print_vertical_detailed).

Pruebas Realizadas

Se elaboraron pruebas en el archivo test_avl.py utilizando pytest.

En total se ejecutaron cinco pruebas que validaron los siguientes aspectos:

Inserción ordenada y mantenimiento del balance.

Eliminación de nodos hoja, con un hijo y con dos hijos.

Verificación del recorrido inorden y del rebalanceo automático.

Todas las pruebas pasaron correctamente, confirmando el funcionamiento esperado del árbol.

Por ejemplo, tras insertar los valores [10, 20, 30, 40, 50, 25], el árbol se equilibró automáticamente, manteniendo el orden inorden [10, 20, 25, 30, 40, 50] con la raíz final en 30.

Conclusión

La versión final del árbol AVL implementada en Lab_AVL_error.py cumple con las propiedades de un árbol balanceado.

Cada vez que se realiza una inserción o eliminación, el algoritmo recalcula las alturas y aplica las rotaciones necesarias (LL, LR, RR o RL) para mantener el equilibrio.

Las pruebas automáticas confirman que el árbol conserva tanto la propiedad de búsqueda como su balance en todos los casos.