

Taller - Árboles AVL

Se nos pide corregir los errores presentes en un código de nombre "Lab_AVL-error.py" cuya función es implementar una clase AVLTree que contenga:

Una clase node auxiliar, funciones auxiliares encargadas: **1)** de calcular la altura, **2)** el factor balance y actualizar la altura de un nodo, **3)** rotaciones simples de izquierda a derecha, **4)** inserción de nuevos valores en el árbol que mantenga el balance AVL del mismo **5)** Eliminación de valores específicos del árbol y que mantengan el balance AVL del mismo **6)** Función de recorrido in-order del árbol y **7)** implementar una función opcional de visualizar/imprimir la estructura del árbol, mostrando los valores, alturas y factores de balance de sus nodos.

A primera instancia al ver el código me enfoqué en destacar con comentarios cada una de las funciones que nos son pedidas y ahí nos damos cuenta de la ausencia del requisito número 5, encargado de la eliminación de valores y el requisito número 6, la función de recorrido in-order del árbol junto al opcional requisito 7. Así que el primer error presente en el código a corregir es la inclusión de dichos elementos, comenzando con la implementación del recorrido in-order, que nos permitirá verificar visualmente si el árbol mantiene la propiedad de ordenamiento después de cada inserción. primero recorrerá el subárbol izquierdo, luego el nodo actual, y finalmente el subárbol derecho. Nos es útil porque en árboles AVL se deberían mostrar los elementos en orden creciente. Al ejecutar nuevamente el código corregido ahora si se puede observar la secuencia de nodos del árbol impresos en consola, demostrando que las inserciones y rotaciones están funcionando como se espera. El recorrido da como resultado: [10, 20, 25, 30, 40, 50], este orden confirma que el árbol está correctamente equilibrado tras las inserciones de los valores [10, 20, 30, 40, 50, 25], ya que los valores se imprimen de menor a mayor sin errores.

Para ver que otros errores se presentan en el código original, ejecutamos el programa, los resultados mostrados en consola al ejecutar el programa son :

```
Insertando valores: [10, 20, 30, 40, 50, 25]
```

```
--- Después de inserciones ---
```

```
** Process exited - Return Code: 0 **
```

```
Press Enter to exit terminal
```

Si bien compila sin errores, esta salida es muy limitada y no incluye ninguna información sobre la estructura del árbol ni verifica si se mantienen las propiedades AVL, con esto se sabe que entonces hay un problema más profundo no evidente a primera vista en las funciones ya encontradas en el árbol; altura, factor balance, rotaciones e inserción de valores. Estas últimas se encontraban defectuosas, ya que las llamadas a las funciones de rotación (`rotate_right` y `rotate_left`) no estaban asignando su resultado a ningún nodo

```
71         if balance > 1 and getBalance(node.left) >= 0:  
72             rotate_right(node) #No se esta guardando el resultado
```

Así que aunque se calculaban las rotaciones necesarias, estas no se estaban aplicando realmente al árbol, provocando que el nodo raíz de cada subárbol no se actualice correctamente tras una rotación.

La solución es sencilla, solamente teniendo que agregar al inicio de la línea un “return” lo cual se garantiza que el nodo actual sea reemplazado por el resultado de la rotación correspondiente, y así el árbol conserva su estructura AVL tras cada inserción.

Al solucionar ese inconveniente, el árbol ya empezó a rotar correctamente y, por tanto, a balancearse tras cada inserción. A partir de allí se incorporó el siguiente requisito faltante:

la eliminación de nodos. Esta función es fundamental para mantener la estructura AVL en casos dinámicos donde los datos pueden ser eliminados. Para su implementación se utilizó un enfoque recursivo, considerando tres casos clásicos: nodo sin hijos, con un hijo y con dos hijos (en cuyo caso se reemplaza por el valor mínimo del subárbol derecho). Después de eliminar un nodo, el árbol es reequilibrado de manera análoga a la inserción, evaluando el balance de cada nodo afectado y realizando las rotaciones necesarias.

Finalmente, como funcionalidad opcional, se agregó una función para imprimir el factor de balance de cada nodo, lo que permite visualizar directamente el estado del equilibrio del árbol tras cada operación. Esta herramienta fue muy útil durante la depuración y para verificar empíricamente que el árbol cumplía las propiedades AVL tras cada inserción o eliminación.