



Tecnológico de Monterrey

Act-2.6 - Árboles binarios de búsqueda AVL

Mauricio Zavala Sánchez - A00837332

19 de octubre del 2023

David Alonso Cantú Delgado

Programación de estructuras de datos y algoritmos fundamentales (Gpo 608)

Árboles Binarios de Búsqueda AVL

Introducción

Los Árboles Binarios de Búsqueda (BST) son una estructura de datos fundamental en la ciencia de la computación que permite almacenar y recuperar datos de manera eficiente. Sin embargo, en un BST común, la altura del árbol puede llegar a ser no balanceada, lo que lleva a peores casos de tiempo de búsqueda. Para abordar este problema, se desarrolló una variante especial de BST llamada "Árbol Binario de Búsqueda Balanceado" o AVL Tree.

Los árboles AVL son una forma de BST que mantienen un equilibrio óptimo en la estructura del árbol, lo que garantiza un rendimiento eficiente en operaciones de búsqueda, inserción y eliminación.

Atributos del ADT (Árbol Binario de Búsqueda AVL)

Un Árbol Binario de Búsqueda AVL tiene los siguientes atributos:

1. Nodo Raíz: Un puntero al nodo raíz del árbol.
2. Nodos: Cada nodo contiene un valor y, opcionalmente, referencias a sus hijos izquierdo y derecho.
3. Factor de Equilibrio (Balance Factor): Cada nodo tiene un valor llamado factor de equilibrio que mide la diferencia entre la altura del subárbol derecho y el subárbol izquierdo. Se expresa como $\text{Balance Factor} = \text{Altura del subárbol derecho} - \text{Altura del subárbol izquierdo}$.
4. Regla de Equilibrio: En un árbol AVL, el factor de equilibrio de cada nodo debe ser -1, 0 o 1 para garantizar el equilibrio.

Métodos Comunes

Los árboles AVL admiten los siguientes métodos comunes:

1. Inserción (Insert): Agregar un nuevo nodo al árbol manteniendo el equilibrio. Si se viola la regla del equilibrio, se realiza una rotación para restaurarla.
2. Eliminación (Delete): Eliminar un nodo del árbol manteniendo el equilibrio. Si se viola la regla del equilibrio, se realiza una rotación para restaurarla.
3. Búsqueda (Search): Buscar un valor específico en el árbol siguiendo el principio de BST.
4. Recorrido (Transversal): Los métodos de recorrido incluye Inorden, Preorden y Postorden para visitar todos los nodos del árbol en un orden específico.
5. Balanceo Automático: Después de las operaciones de inserción y eliminación, el árbol AVL se reequilibra automáticamente.

Mantenimiento del Balance

1. Inserción de un Nuevo Nodo

Cuando se agrega un nuevo nodo a un árbol AVL, se sigue este proceso:

1. Realiza la inserción como en un BST normal.
2. Actualiza el factor de equilibrio de todos los nodos desde el nodo recién insertado hasta la raíz.
3. Si se viola la regla de equilibrio (-1, 0, 1) en algún nodo, se realiza una de las cuatro rotaciones posibles (RR, LL, RL, LR) para restaurar el equilibrio.
4. Repite el proceso hasta llegar a la raíz.

2. Eliminación de un Nodo

Cuando se elimina un nodo de un árbol AVL, se sigue este proceso:

1. Realiza la eliminación como en un BST normal.
2. Actualiza el factor de equilibrio de todos los nodos desde el nodo eliminado hasta la raíz.
3. Si se viola la regla de equilibrio (-1, 0, 1) en algún nodo, se realiza una de las cuatro rotaciones posibles (RR, LL, RL, LR) para restaurar el equilibrio.
4. Repite el proceso hasta llegar a la raíz.

Uso Aplicativo

Los árboles AVL se utilizan en aplicaciones donde se requiere un rendimiento de búsqueda eficiente y se necesita garantizar tiempos de búsqueda predecibles. Algunos ejemplos de uso aplicativo son:

1. Bases de Datos: Se utilizan en sistemas de bases de datos para indexar y buscar registros de manera eficiente.
2. Editores de texto avanzados: Para implementar la función "Buscar" de manera eficiente.
3. Compiladores: Se emplean para optimizar la búsqueda de variables y funciones en el código fuente.
4. Sistemas de Archivos: En algunos sistemas de archivos, se usan árboles AVL para acelerar la búsqueda de archivos o directorios.
5. Redes y enrutamiento: En la gestión de rutas y búsqueda de direcciones IP.