Universidad Cenfotec

Materia:

Estructuras de Datos.

Proyecto:

Implementación de la Red de la empresa

Profesor:

Romario Salas Cerdas.

Estudiantes:

Emanuel Wong Gamboa

Introduccion:

Los árboles AVL son una de las estructuras de datos auto balanceadas más comunes en la informática actual. Fueron sugeridos por Adelson-Velskii y Landis en 1962, con la intención de solucionar un inconveniente inherente a los árboles binarios de búsqueda clásicos, su propensión a desbalancear tras inserciones o eliminaciones repetidas de nodos. Cuando esto sucede, el árbol puede transformarse en una estructura lineal, perdiendo totalmente la eficiencia típica de las operaciones de búsqueda, inserción y eliminación.

Los árboles AVL solucionan este inconveniente mediante el estricto mantenimiento de una propiedad de equilibrio que asegura que la diferencia entre las alturas de los subárboles izquierdo y derecho de cada nodo sea siempre −1, 0 o +1.

Este monitoreo del equilibrio transforma a los árboles AVL en una estructura de datos particularmente útil en situaciones donde las operaciones deben ser eficaces incluso en el escenario más desfavorable. La habilidad de autocorrección a través de rotaciones permite que la estructura se restructure automáticamente para mantener un tiempo de ejecución óptimo de O(log n). Debido a estas características, los árboles AVL son utilizados extensamente en sistemas de bases de datos, procesamiento de información en tiempo real, gestión de grandes volúmenes de datos dinámicos y varios componentes críticos de software.

Este documento presenta un análisis exhaustivo del cálculo del factor de equilibrio en los nodos de un árbol AVL y de los algoritmos de rotación requeridos para corregir desbalances. Igualmente, se presentan representaciones conceptuales de estas operaciones y se detallan aplicaciones reales donde los árboles AVL son la estructura de datos más idónea, justificando la relevancia de su uso en esos casos

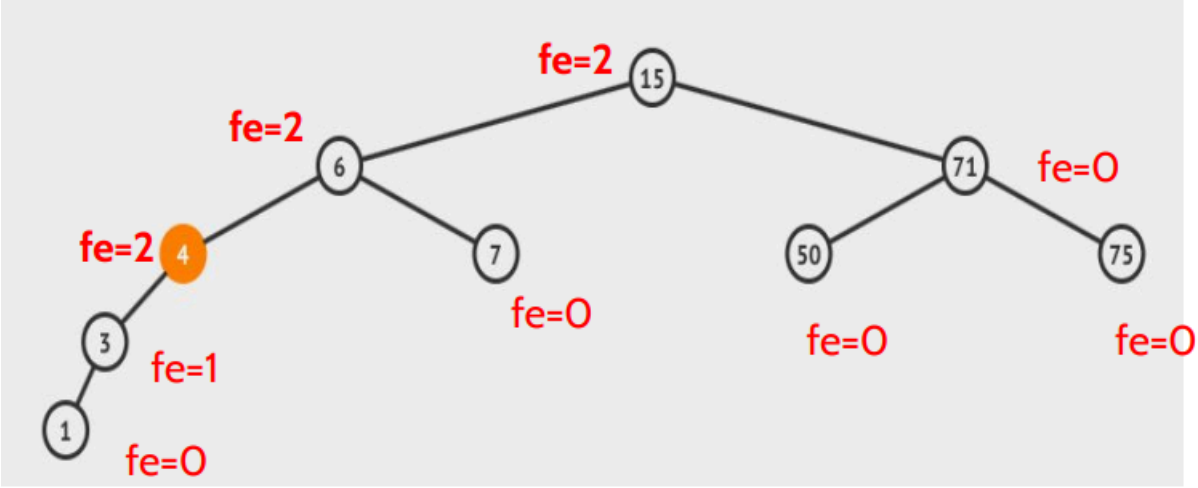
Evaluación del factor de balance en árboles AVL

El factor de balance (FB) es el elemento central que permite determinar si un árbol AVL mantiene sus propiedades de equilibrio. Para cada nodo del árbol, el factor de balance se define como la diferencia entre la altura del subárbol izquierdo y la altura del subárbol derecho

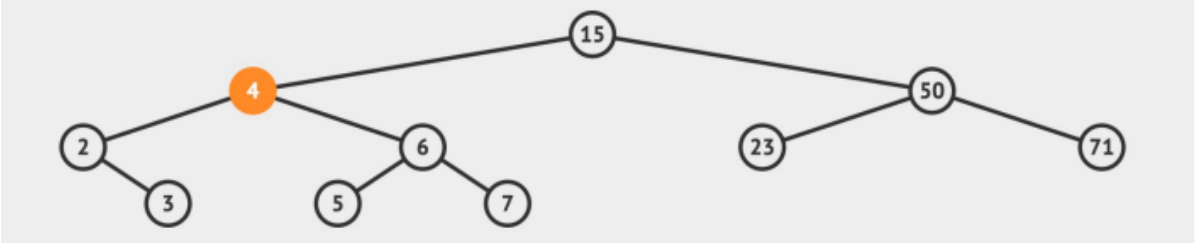
FB=altura(izquierdo)−altura(derecho)FB = altura(izquierdo) - altura(derecho)FB=altura(izquierdo)−altura(derecho)

Para que un nodo cumpla con las condiciones de balance de un árbol AVL, su factor de balance debe encontrarse estrictamente dentro del rango permitido **−1, 0 o +1**. Si un nodo presenta un factor de balance fuera de este intervalo, se considera que el árbol está desbalanceado y que requiere una o varias rotaciones para restaurar el equilibrio.

Este está desequilibrado



Después del auto balance



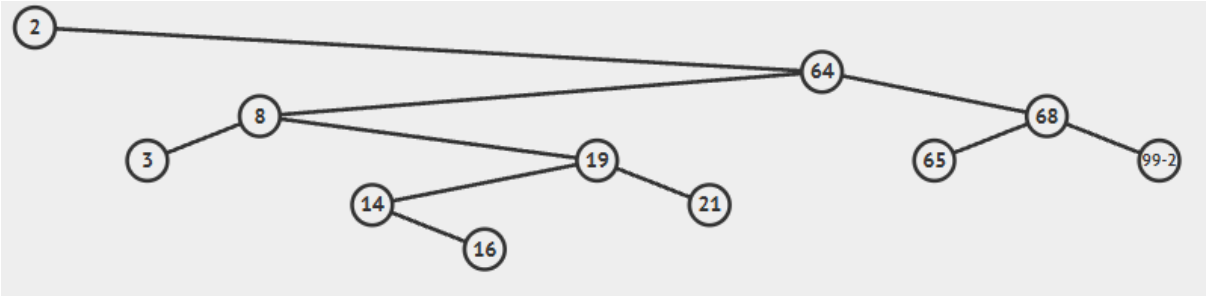
La altura de un nodo se define como la longitud del camino más largo desde ese nodo hasta una hoja. Por convención, un nodo nulo posee altura −1, mientras que una hoja tiene altura 0. La altura de cualquier nodo no vacío puede calcularse mediante

altura(nodo)=1+max(altura(izquierdo),altura(derecho))altura(nodo) = 1 + \max(altura(izquierdo), altura(derecho))altura(nodo)=1+max(altura(izquierdo),altura(derecho))

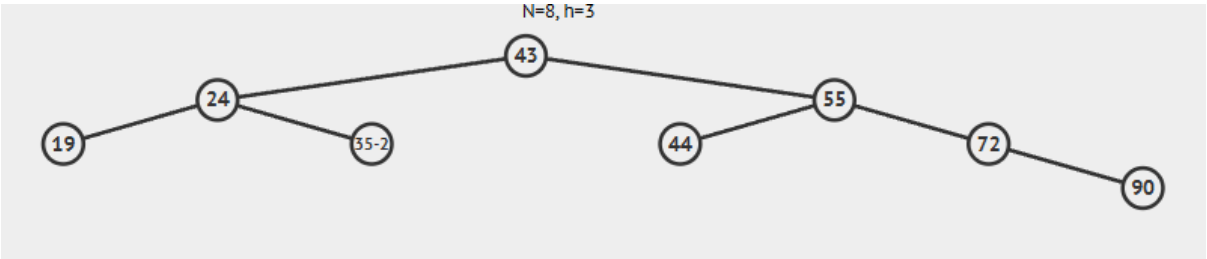
Después de cada operación de inserción o eliminación, las alturas de los nodos afectados deben calcularse de manera ascendente hacia la raíz. Esto permite obtener factores de balance actualizados y detectar posibles anomalías en la estructura. La evaluación se realiza nodo por nodo mientras se retorna en la recursión, o bien utilizando un proceso iterativo según la implementación.

Este cálculo es crucial porque determina si la posterior aplicación de una rotación será necesaria. De hecho, los árboles AVL se distinguen por hacer explícito y sistemático este análisis después de cada modificación de la estructura. En términos operativos, la evaluación del factor de balance asegura que las operaciones de búsqueda, inserción y eliminación mantengan su eficiencia logarítmica, previniendo la degradación típica de los árboles binarios no balanceados.

Arbol binario



AVL



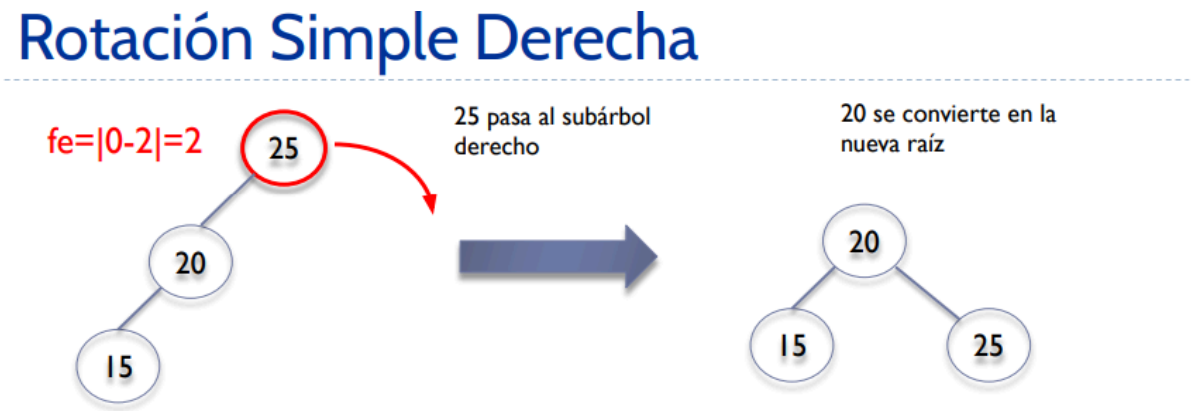
Algoritmos de rotación para la corrección de desbalances

Cuando el factor de balance de un nodo es menor que −1 o mayor que +1, el árbol AVL debe aplicar uno de los cuatro algoritmos de rotación disponibles para devolver el equilibrio. Las rotaciones reorganizan la estructura del árbol sin violar las propiedades del árbol binario de búsqueda. Los cuatro tipos de rotaciones responden a las combinaciones posibles de desbalance y al lugar donde se produjo la inserción o eliminación.

Rotación simple a la derecha

Este procedimiento se aplica cuando un nodo presenta un desbalance positivo (FB > 1) y la inserción ocurrió en el subárbol izquierdo del hijo izquierdo. En este caso, el hijo izquierdo asciende a la posición del nodo desbalanceado, y este último se convierte en el hijo derecho del nuevo padre. Esta rotación reduce la altura del subárbol izquierdo y corrige un desbalance de tipo izquierda–izquierda (LL).

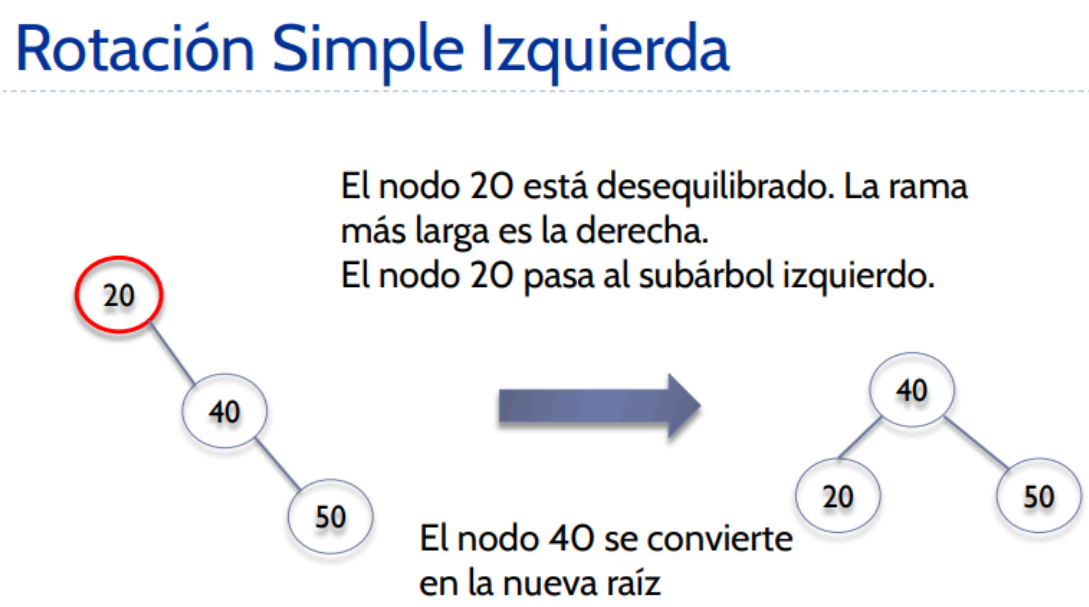
La rotación simple a la derecha es la más intuitiva de todas, pues su efecto es “inclinar” el árbol hacia la derecha para compensar una sobrecarga en el lado izquierdo. Es eficiente, simple y mantiene intactas las propiedades del árbol de búsqueda.



Rotación simple a la izquierda

Es el caso opuesto al anterior. Se utiliza cuando el nodo tiene FB < −1 y la inserción ocurrió en el subárbol derecho del hijo derecho. La rotación consiste en promover al hijo derecho del nodo desbalanceado y desplazar el nodo original hacia la izquierda. Esta operación corrige desbalances de tipo derecha–derecha (RR).

Conceptualmente, la rotación simple a la izquierda “levanta” el lado derecho del árbol y equilibra la estructura reduciendo la altura del subárbol derecho.



Rotación doble izquierda–derecha

Esta rotación se aplica cuando el nodo está desbalanceado hacia la izquierda pero la inserción ocurrió en el subárbol derecho del hijo izquierdo, es decir, en una configuración izquierda–derecha (LR). Para corregirla

1. Se realiza primero una rotación simple a la izquierda sobre el hijo izquierdo del nodo desbalanceado.

2. Luego se aplica una rotación simple a la derecha sobre el nodo desbalanceado.

Este proceso corrige la doble inclinación, ajustando el árbol a una estructura balanceada.

Rotación doble derecha–izquierda

De forma simétrica al caso anterior, esta rotación se utiliza cuando ocurre un desbalance hacia la derecha pero la inserción sucede en el subárbol izquierdo del hijo derecho (casó derecha–izquierda, RI). El procedimiento consiste en:

1. Realizar una rotación simple a la derecha sobre el hijo derecho del nodo desbalanceado.

2. Posteriormente aplicar una rotación simple a la izquierda sobre el nodo desbalanceado.

Las rotaciones dobles resuelven los escenarios más complejos, donde la estructura presenta inclinaciones cruzadas. Su correcta implementación es indispensable para garantizar la integridad del árbol AVL.

Aplicaciones de los árboles AVL

Bases de datos e índices de búsqueda

Los AVL permiten mantener colecciones de datos dinámicos ordenados con tiempos de inserción, eliminación y búsqueda garantizados en O(log n). Esto es fundamental para índices secundarios, catálogos y sistemas que deben responder rápidamente incluso bajo carga.

Sistemas de archivos y metadatos

En estructuras donde la organización jerárquica cambia de manera frecuente, como los sistemas de archivos, la capacidad de balance automático garantiza accesos eficientes y uniformes.

Motores de búsqueda y recuperación de información

El mantenimiento del orden y la alta velocidad de acceso permiten implementar listas de términos, tablas de símbolos y estructuras para autocompletado.

Sistemas en tiempo real

En aplicaciones que no toleran variabilidad en el peor caso, como control de dispositivos, compiladores o routers, los AVL proporcionan tiempos de procesamiento consistentes.

En todas estas situaciones, la elección de árboles AVL asegura confiabilidad y eficiencia debido a su balanceo automático continuo.

Biografías

Adelson-Velskii, G. M., Landis, E. M. (1962).

An algorithm for the organization of information.
Soviet Mathematics Doklady, 3, 1259–1262.
(Documento original donde se introducen los árboles AVL.)

Weiss, M. A. (2014).

Data Structures and Algorithm Analysis in Java (3rd ed.). Pearson.
(Un libro estándar que explica AVL, factores de balance y rotaciones.)

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009).

Introduction to Algorithms (3rd ed.). MIT Press.
(Incluye análisis de árboles balanceados, especialmente AVL y rotaciones.)

Goodrich, M. T., Tamassia, R., & Goldwasser, M. (2014).

Data Structures & Algorithms in Java (6th ed.). Wiley.
(Referencia clara para implementaciones de AVL en Java.)

Shaffer, C. A. (2013).

Data Structures and Algorithm Analysis.
(Contiene explicaciones detalladas de rotaciones y operaciones sobre ABB/AVL.)

VisuAlgo. (2024). Binary Search Tree & AVL rotations.

https://visualgo.net/en/bst
(Utilizado para describir paso a paso rotaciones, coincidiendo con los ejemplos visuales.)

OpenCourseWare – Grado en Ingeniería Informática. (s.f.).

TEMA 5: Árboles – 5.3 Árboles AVL.
Estructura de Datos y Algoritmos.

Notas sobre esta fuente:

- Aporta la explicación del factor de equilibrio (pág. 7)
- Las condiciones de un AVL (pág. 8),
- Ejemplos de nodos desbalanceados (pág. 9),
- Explicaciones gráficas de todas las rotaciones:
    - Rotación simple derecha (pág. 13–22)
    - Rotación simple izquierda (pág. 23–31)
    - Doble izquierda–derecha (pág. 32–41)
    - Doble derecha–izquierda (pág. 42–52)

OpenAI. (2025). *ChatGPT* (GPT-5.1) [Modelo de lenguaje de inteligencia artificial]. https://chat.openai.com/