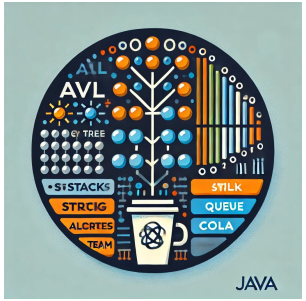


CLASE 8

Ing. Silvestre Alejandro
Informática III
IUA - 2024

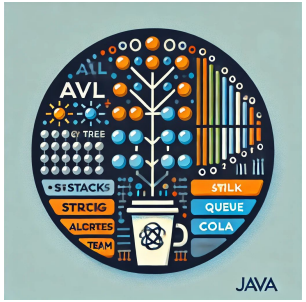
Objetivos del día

1. Cronología de las próximas clases.
2. Repaso Árbol AVL.
3. Práctico ABL.

Cronología de las próximas clases

3/10/2024	**Árbol AVL**
10/10/2024	**Árbol Rojinegro**
17/10/2024	**Montículo Binario**
31/10/2024 al 7/11/2024	**Proyecto Consultoría**
14/11/2024	**Segundo Parcial** Presentación Oral del proyecto
21/11/2024	**Recuperatorio**

ÁRBOL AVL


AVL
AVL
STACK
STRCIG
ALCRTES
TEAM
STILK
QUEUE
COLA
JAVA


1941

Árbol AVL

Los Árboles AVL son una estructura de datos de tipo árbol binario de búsqueda que garantiza un balanceo óptimo, lo que mejora el rendimiento de las operaciones.

Conceptos Fundamentales

Árbol Binario de Búsqueda (BST)

- Un árbol binario de búsqueda es una estructura de datos en la que cada nodo tiene, a lo sumo, dos hijos (izquierdo y derecho), y para cada nodo, todos los nodos del subárbol izquierdo tienen un valor menor que el nodo y todos los nodos del subárbol derecho tienen un valor mayor.

Altura de un Nodo

- La altura de un nodo se refiere a la longitud del camino más largo desde ese nodo hasta una hoja. Es decir, el número de aristas en el camino más largo desde el nodo hasta una hoja.
- Puede haber nodos con altura 0 (hojas) y nodos con altura mayor.

Árbol Balanceado

- Un árbol se considera balanceado si la diferencia de altura entre el subárbol izquierdo y el subárbol derecho de cada nodo es como máximo 1.
- Ventajas de los árboles balanceados: Garantizan un tiempo de ejecución eficiente para operaciones como búsqueda, inserción y eliminación.

Propiedades de un Árbol AVL

Propiedad de Balance

- En un Árbol AVL, la diferencia de altura entre el subárbol izquierdo y el subárbol derecho de cada nodo debe ser como máximo 1. Esto significa que ningún nodo puede tener un desequilibrio mayor que 1 en términos de altura. Si se viola esta propiedad, se deben realizar rotaciones para restaurar el equilibrio.

Diferencia de Altura entre Subárboles

- Para cada nodo, se calcula la diferencia de altura entre el subárbol izquierdo y el subárbol derecho (altura(izquierdo) - altura(derecho)). Si esta diferencia es mayor que 1 o menor que -1, el árbol no cumple con la propiedad de balance y se debe realizar una rotación.

Ventajas del Árbol AVL

- Tiempo de ejecución predecible: Las operaciones de búsqueda, inserción y eliminación tienen un tiempo de ejecución garantizado de O(log n), lo que significa que el rendimiento es consistente y no está sujeto a peores casos.
- Ideal para aplicaciones que requieren un rendimiento constante en operaciones de búsqueda y manipulación de datos.

Desventajas del Árbol AVL

- Mayor complejidad en la implementación: Mantener la propiedad de balance durante las operaciones puede requerir rotaciones, lo que puede aumentar la complejidad de implementación en comparación con árboles binarios de búsqueda simples.

Casos de Uso de los Árboles AVL

Bases de Datos
Los Árboles AVL son ampliamente utilizados en sistemas de gestión de bases de datos para indexar y buscar registros de manera eficiente. Garantizan tiempos de búsqueda y recuperación predecibles, lo que es crucial en aplicaciones que manejan grandes volúmenes de datos.

Sistemas de Búsqueda en Tiempo Real
Aplicaciones que requieren búsquedas en tiempo real, como motores de búsqueda, se benefician de los Árboles AVL. Garantizan un tiempo de búsqueda constante y rápido incluso con grandes conjuntos de datos.

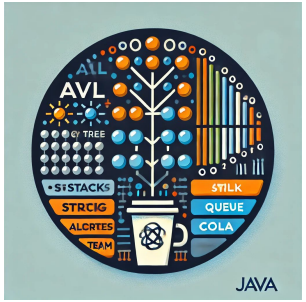
Sistemas de Información Geográfica (SIG)
En SIG, los Árboles AVL pueden utilizarse para indexar datos espaciales, como ubicaciones geográficas. Esto permite una rápida recuperación de información basada en coordenadas geográficas.

Sistemas de Archivos y Sistemas Operativo
Algunos sistemas de archivos y sistemas operativos utilizan Árboles AVL para mantener la organización y búsqueda eficiente de archivos y directorios. Esto es esencial para optimizar operaciones de lectura y escritura.

Bases de Datos de Grafos
En bases de datos de grafos, los Árboles AVL pueden ser utilizados como índices para acelerar las búsquedas en la estructura de grafo, especialmente cuando se necesita encontrar rápidamente nodos o relaciones específicas.

Algoritmos de Juegos y Simulaciones
En aplicaciones de juegos y simulaciones, los Árboles AVL pueden utilizarse para organizar y buscar datos como objetos, personajes o eventos en el mundo del juego. Esto mejora la eficiencia de procesos como colisiones y detección de eventos.

ÁRBOL AVL EN JAVA



Clase Nodo

```
class Nodo {
    int valor, altura; //diferencia de alturas entre el subárbol izquierdo y el subárbol derecho de un nodo
    Nodo izquierdo, derecho;
    public Nodo(int valor) {
        this.valor = valor;
        this.altura = 0;
        this.izquierdo = this.derecho = null;
    }
}
```

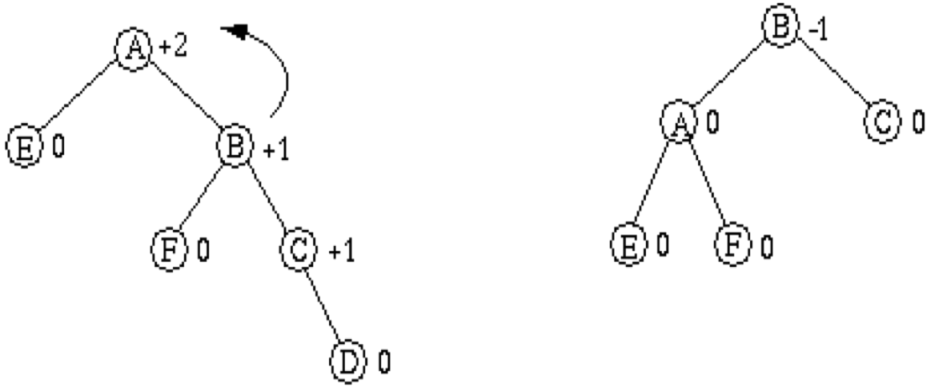
Método Altura

```
// Método para obtener la altura de un nodo
int altura(Nodo nodo) {
    if (nodo == null) return 0;
    return nodo.altura;
}

// Método para actualizar la altura de un nodo
void actualizarAltura(Nodo nodo) {
    nodo.altura = 1 + Math.max(altura(nodo.izquierdo), altura(nodo.derecho));
}
```

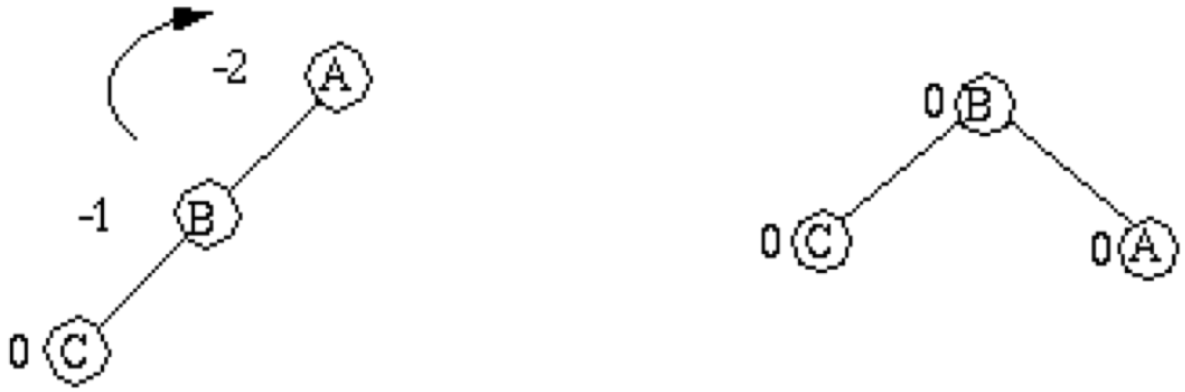
Método Rotación Izquierda simple

Si está desequilibrado a la **izquierda** y su hijo derecho tiene el mismo signo **(+)** hacemos rotación sencilla izquierda.



Método Rotación Derecha simple

Si está desequilibrado a la derecha y su hijo izquierdo tiene el mismo signo (-) hacemos rotación sencilla derecha.



Método Insertar

PASOS

1. Insertar el nodo.
2. Actualizar las alturas de los nodos.
3. Rotar en caso de que sea necesario.

Práctico 8