Relatório Árvores AVL

Nome: Marcelo Augusto Corrêa Cesario Vasques de Lima

**1. História da Criação do Algoritmo (Contexto Histórico)**

* As árvores AVL foram criadas em **1962** por dois matemáticos soviéticos, **Georgy Adelson-Velsky** e **Evgenii Landis**, daí o nome AVL.
* O objetivo era desenvolver uma estrutura de dados que mantivesse a árvore de pesquisa binária equilibrada para melhorar a eficiência nas operações de pesquisa, inserção e remoção.
* A sua criação surgiu da necessidade de otimizar as árvores binárias, que, sem equilíbrio, podem se transformar em listas ligadas, prejudicando a eficiência.

**2. Conceitos Principais da Árvore AVL**

***Árvore Binária de Busca:***

Uma árvore em que, para cada nó, os valores à esquerda são menores e os valores à direita são maiores.

***Balanceamento:***

Uma árvore AVL é uma árvore binária de busca que mantém o seu fator de balanceamento controlado.

***Fator de Balanceamento (FB):***

É a diferença entre a **altura da subárvore** à esquerda e à direita de um nó.

**FB** = Altura (subárvore esquerda) – Altura (subárvore direita)

Para ser AVL, o fator de balanceamento de todos os nós deve estar entre -1 e 1.

***Rotação:***

Quando um nó fica desequilibrado (FB fora de -1 e 1), a árvore é reequilibrada por meio de rotações:

Rotação Simples à Direita.

Rotação Simples à Esquerda.

Rotação Dupla à Direita (Esquerda-Direita).

Rotação Dupla à Esquerda (Direita-Esquerda).

**3. Quando o Uso é Mais Eficiente**

* As árvores AVL são mais eficientes quando é necessário:
  + **Pesquisa rápida** de elementos, mantendo o tempo de busca em **O(log n)**, pois a árvore está sempre balanceada.
  + **Inserção e remoção frequente** de dados, especialmente quando o volume de dados pode causar desequilíbrios em outras estruturas, como árvores binárias de busca simples.
  + Em cenários onde a **altura da árvore** tem impacto direto no desempenho (por exemplo, bancos de dados ou sistemas com grandes conjuntos de dados dinâmicos).

**4. Como Funciona a Lógica da Árvore AVL**

* A **inserção** e a **remoção** de nós seguem as mesmas regras de uma árvore binária de busca normal.
* **Após cada operação (inserção ou remoção)**:
  + Calcula-se o fator de balanceamento de cada nó, começando do nó modificado até à raiz.
  + Se um nó estiver **desequilibrado** (FB fora de -1 a 1), aplica-se uma **rotação** (ou combinação de rotações) para restaurar o equilíbrio.
  + As rotações permitem que a árvore retorne ao estado de equilíbrio com uma altura mínima, garantindo que as operações de busca continuem eficientes.
* **Complexidade**:
  + As operações de inserção, remoção e busca mantêm uma complexidade de **O(log n)**, graças ao controle da altura da árvore.

**Conclusão**

* As árvores AVL são uma solução eficiente para garantir que a estrutura de dados se mantenha balanceada, permitindo operações rápidas e consistentes. São particularmente úteis em aplicações que exigem um elevado número de inserções e remoções, sem comprometer a eficiência das pesquisas.