

AVL resumo simples:

Introdução:

A árvore AVL é uma árvore binária semibalanceada, ela entra no nosso contexto porque uma árvore binária de busca balanceada, se sofrer muitas inserções e deleções ficará ineficiente.

Na árvore balanceada temos um critério de balanceamento relaxado, mais liberdade na definição de algoritmos, algoritmos mais eficientes etc.

Rotações:

Para manter a eficiência da árvore iremos usar rotações para manter o equilíbrio. A ideia por trás das rotações é reorganizar os nós na árvore para que a altura das duas subárvores de um nó permaneça aproximadamente a mesma.

Existem dois tipos de rotações nas árvores AVL: Rotação à Esquerda e Rotação à Direita.

Rotação Esquerda:

Considere uma árvore onde a subárvore direita de um nó é muito alta. Isso pode ser corrigido girando o nó para a esquerda. A rotação à esquerda pega o filho direito do nó e o torna a nova raiz da subárvore, com o nó original tornando-se o filho esquerdo da nova raiz.

Rotação direita:

Considere uma árvore onde a subárvore esquerda de um nó é muito alta. Isso pode ser corrigido girando o nó para a direita. A rotação à direita pega o filho esquerdo do nó e o torna a nova raiz da subárvore, com o nó original tornando-se o filho direito da nova raiz.

Conclusão:

Vamos garantir que a árvore mantenha o equilíbrio verificando a altura das subárvores após cada inserção ou exclusão e aplicando rotações, se necessário, para manter a diferença de altura entre as subárvores esquerda e direita em no máximo 1.

Em geral, as rotações nas árvores AVL levam tempo $O(1)$ e são usadas para garantir que a altura da árvore seja sempre logarítmica no número de elementos, o que resulta em operações rápidas de pesquisa, inserção e exclusão.