Estruturas de Dados

Aula 11 – Árvores AVL

Prof. Ana Carolina Sokolonski

Bacharelado em Sistemas de Informação
Instituto Federal da Bahia – Campus Feira de Santana

2026

Agenda

Motivação

Definição

Balanceamento

Trabalho

Referências

Motivação

Por que precisamos de AVL?

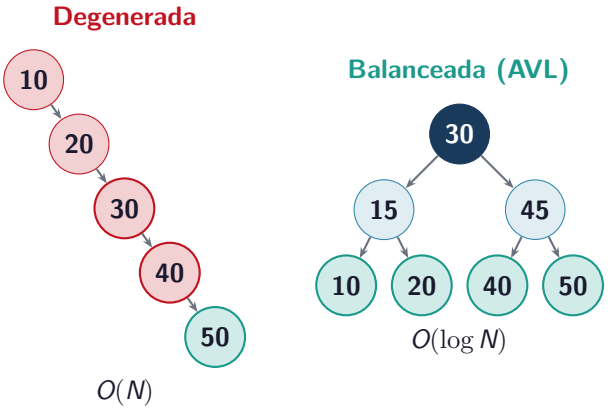
Árvore Degenerada — O Problema

▪ Árvore Degenerada

Quando os elementos são inseridos em ordem crescente ou decrescente em uma ABB, a árvore degenera em uma **lista encadeada**. A busca passa de $O(\log N)$ para $O(N)$!

▪ Solução: Árvore AVL

Mantém a árvore **balanceada** após cada inserção/remoção, garantindo busca em $O(\log N)$ sempre.



Árvores AVL

Adelson-Velsky e Landis, 1962

O que é uma Árvore AVL?

▪ Definição

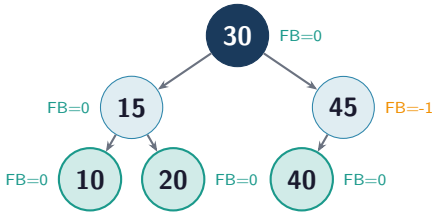
ABB **balanceada**: para qualquer nó u, alturas das subárvores diferem em no máximo **1**. [Cormen et al. 2009] Nome: **A**delson-**V**elsky e **L**andis (1962).

▪ Complexidade

Busca, Inserção e Remoção: $O(\log n)$

▪ Fator de Balanço (FB)

$\text{FB}(u) = h_d(u) - h_e(u)$
Nó **regulado**: $\text{FB}(u) \in \{-1,\ 0,\ +1\}$
$|\text{FB}| > 1 \Rightarrow$ árvore **não é AVL**

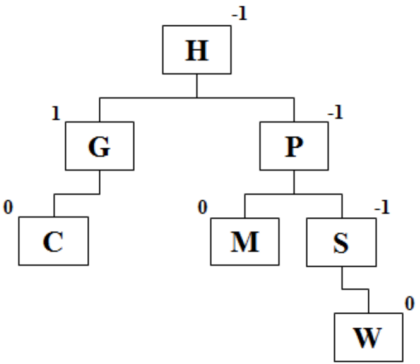


- FB=$\{-1, 0, +1\}$: AVL
- $|\text{FB}| > 1$: não AVL

+1: subárvore esquerda direita

0: subárvore esquerda

-1: subárvore direita ma esquerda



Estrutura do Nó AVL

Campos do nó

Além dos campos de uma ABB comum, o nó AVL armazena o **fator de balanço** (ou a **altura**) para que as rotações sejam realizadas de forma eficiente.

Struct típica em C

```
typedef struct No {
  int dado;
  int fatorBalanco;
  struct No *esq;
  struct No *dir;
} No;
```

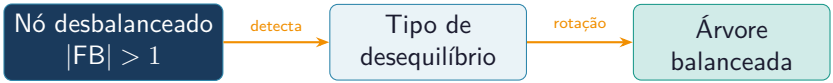
```
typedef struct noArvoreAVL{
  int chave;
  int fatorBalanco;
  struct noArvoreAVL *filhoDir;
  struct noArvoreAVL *filhoEsq;
}NO;
```

Balanceamento

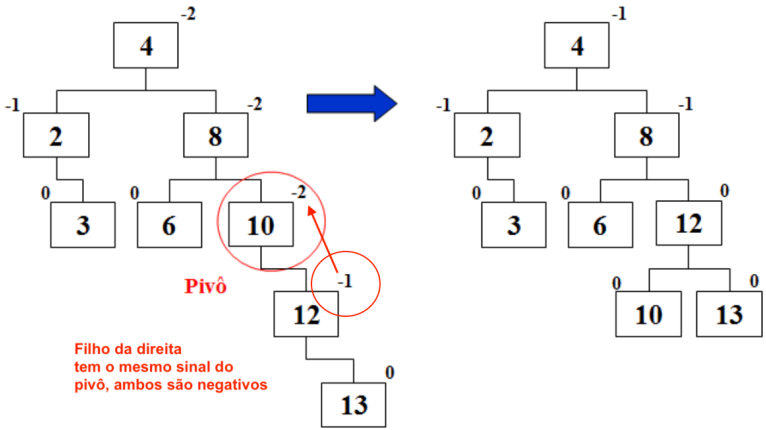
Rotações Simples e Duplas

Os 4 Tipos de Rotação

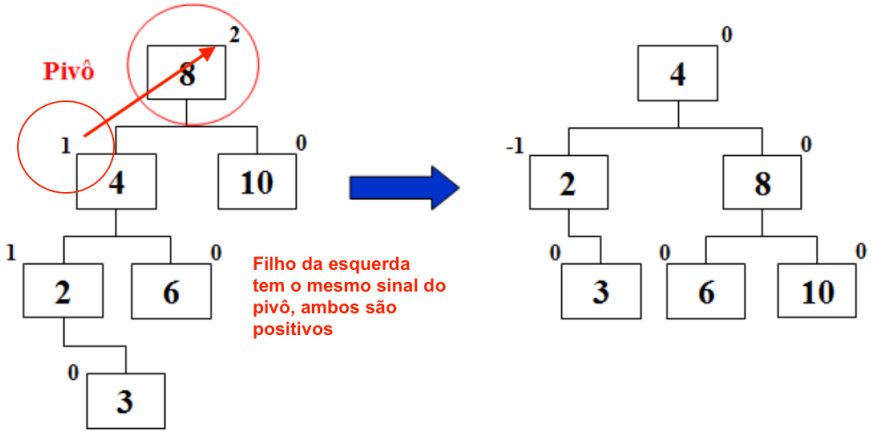
Rotação	Quando usar	FB do nó desbalanceado
Simples Esquerda	Subárvore **direita** muito alta	$+2$ e filho dir com FB ≥ 0
Simples Direita	Subárvore **esquerda** muito alta	-2 e filho esq com FB ≤ 0
Dupla Esquerda	Filho dir com subárvore **esq** alta	$+2$ e filho dir com FB < 0
Dupla Direita	Filho esq com subárvore **dir** alta	-2 e filho esq com FB > 0



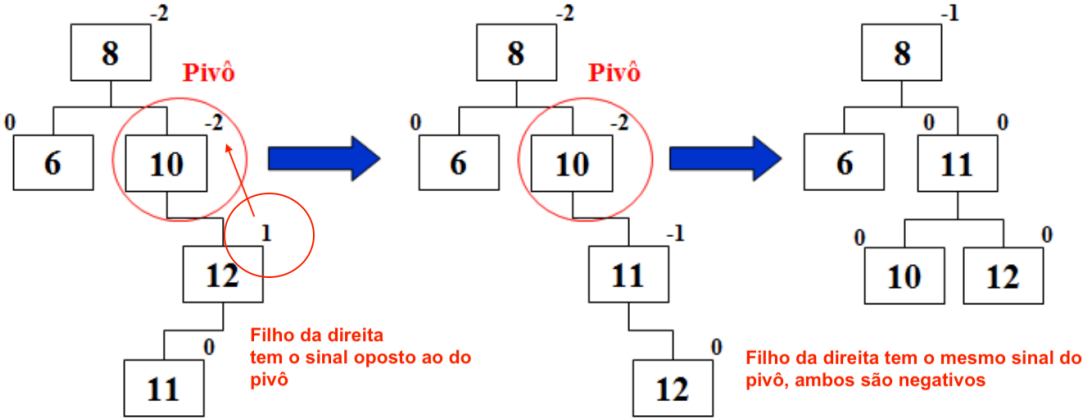
Rotação Simples à Esquerda



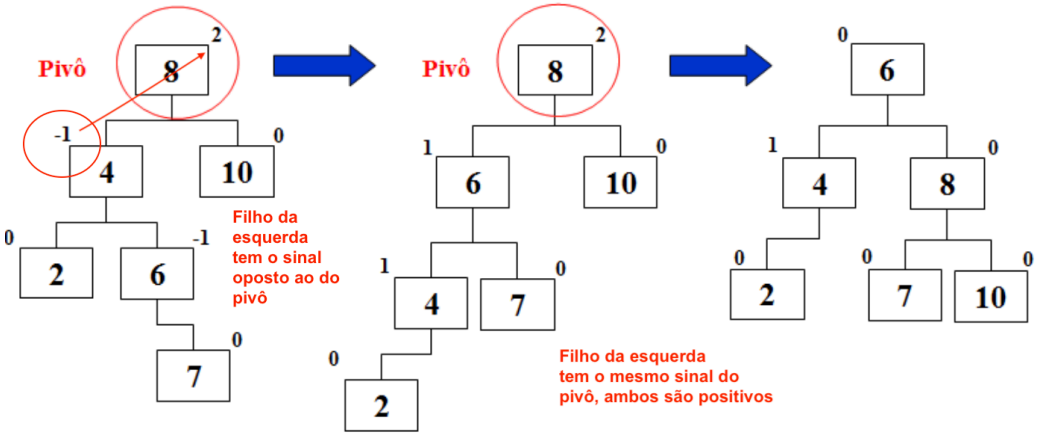
Rotação Simples à Direita



Rotação Dupla à Esquerda



Rotação Dupla à Direita



Exercício

Implementação da Árvore AVL

Implementação — Menu de Operações

Implemente a Árvore AVL com o seguinte menu

1. Inserir elemento na Árvore AVL
2. Mostrar em **Pré-ordem**
3. Mostrar em **Pós-ordem**
4. Mostrar **Em-Ordem**
5. **Desenhar** a Árvore AVL da raiz às folhas
6. **Remover** elemento da Árvore AVL
7. **Buscar** elemento na Árvore AVL
8. Fechar

Atenção

Toda inserção/remoção deve verificar o **fator de balanço** e aplicar a rotação adequada quando necessário.

Referências I

CORMEN, T. H. et al. *Introduction to Algorithms*. 2nd. ed. [S.l.]: The MIT Press, 2009. ISBN 0262032937.