



Árvores Balanceadas (AVL)



Roteiro

- Contextualização
- Árvores Balanceadas (AVL)
- Operações de Balanceamento



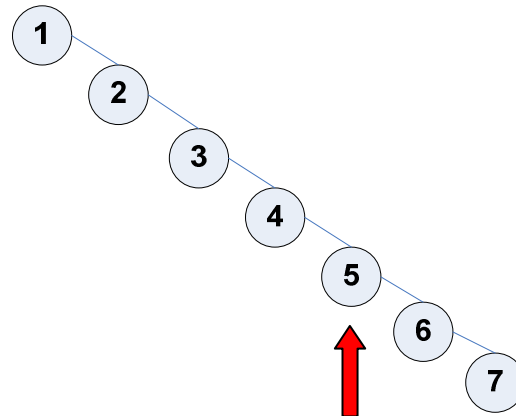
Roteiro

- **Contextualização**
- Árvores Balanceadas (AVL)
- Operações de Balanceamento

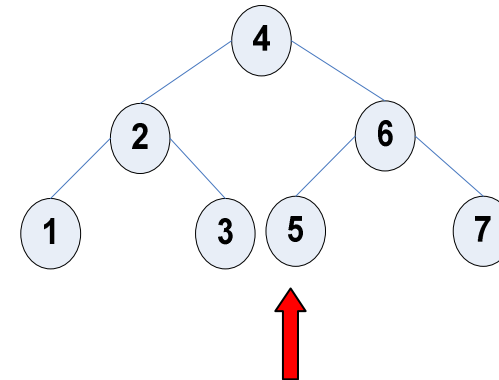
Contextualização

- As AB estudadas têm uma **séria desvantagem** que pode afetar o tempo necessário para recuperar um item armazenado.
- A desvantagem é que o desempenho da AB **depende da ordem em que os elementos são inseridos**.

1, 2, 3, 4, 5, 6, 7



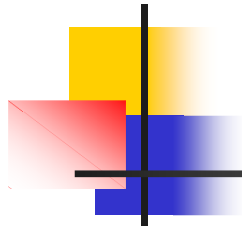
4, 6, 2, 5, 1, 7, 3





Contextualização

- Idealmente, deseja-se que a árvore esteja **balanceada**, para qualquer nó **p** da árvore.
- Como saber se a árvore está **balanceada** ?
- Para cada nó **p** da árvore a altura da sua **sae** é **aproximadamente igual** à altura da sua **sad**.



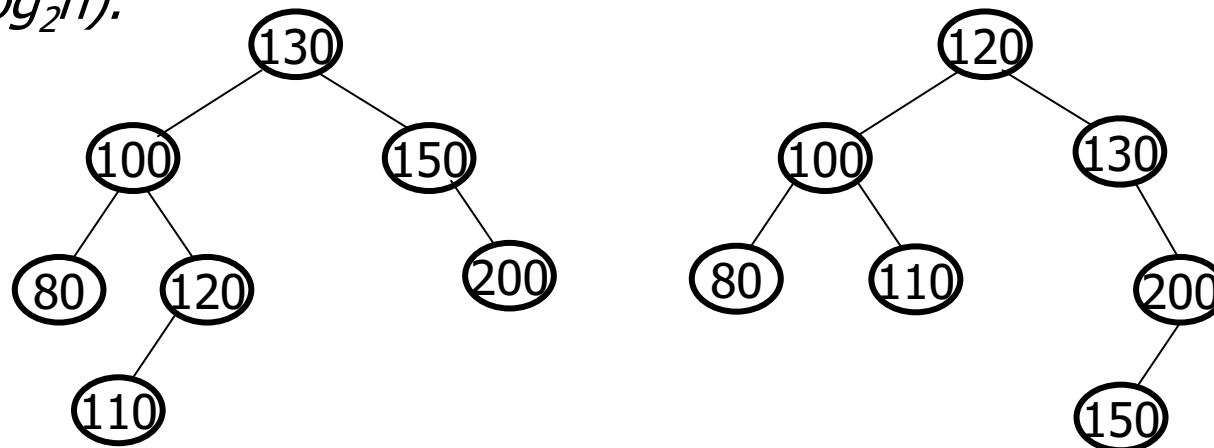
Roteiro

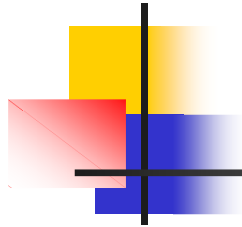
- Contextualização
- **Árvores Balanceadas (AVL)**
- Operações de Balanceamento



Árvores Balanceadas (AVL)

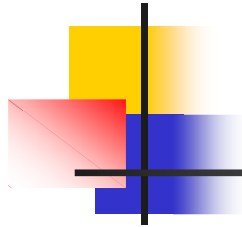
- O nome AVL vem de seus criadores Adelson Velsky e Landis (1962).
- Uma árvore binária de pesquisa **T** é denominada **AVL** se:
 - Para todos nós de **T**, as alturas de suas duas sub-árvores diferem **no máximo de uma unidade**.
- Operações de consulta, inserção e remoção de nós tem custo $O(\log_2 n)$.





Como reconhecer uma árvore desbalanceada? (1/2)

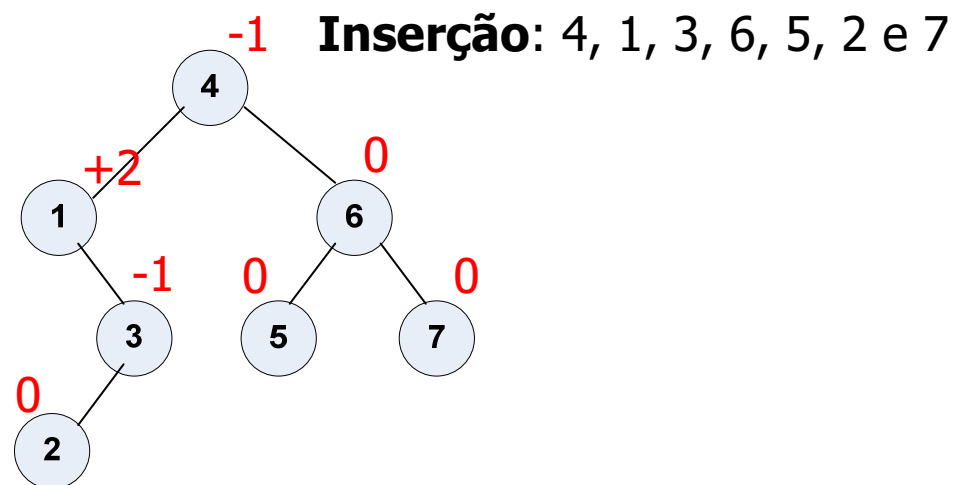
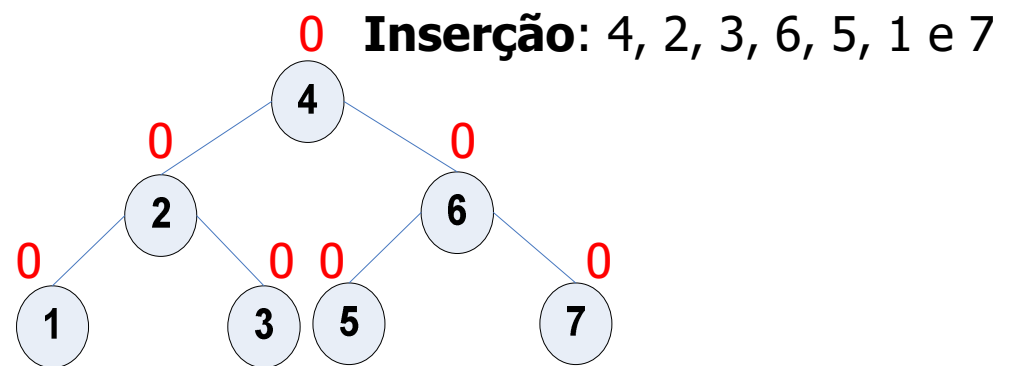
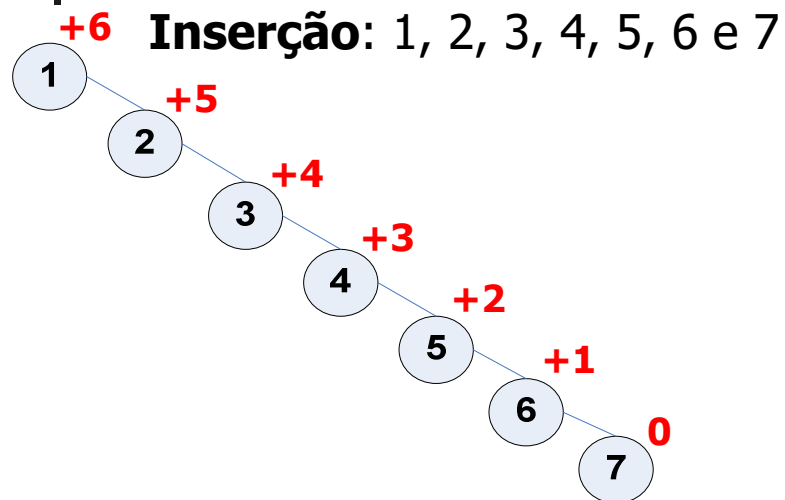
- Como saber se a árvore está **desbalanceada** ?
 - Verificando se existe algum nodo “desregulado”.
- Como saber se um nodo está **desregulado** ?
 - Subtraindo-se as alturas das suas sub-árvores.
- Por questões de **eficiência**, estas diferenças são pré-calculadas e armazenadas nos nós correspondentes, sendo atualizadas durante as operações.



Como reconhecer uma árvore desbalanceada? (2/2)

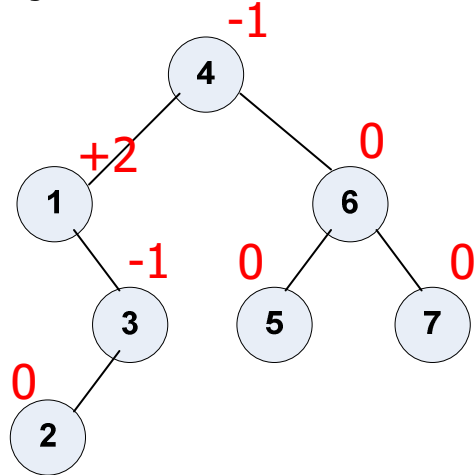
- Possíveis valores de diferença para cada nó em uma árvore balanceada: **-1, 0, 1**.
- Fator de Balanceamento (FB) de cada nó da árvore
 - **FB(p)** = $h(\text{sad}(p)) - h(\text{sae}(p))$

Exemplos de cálculos de FB

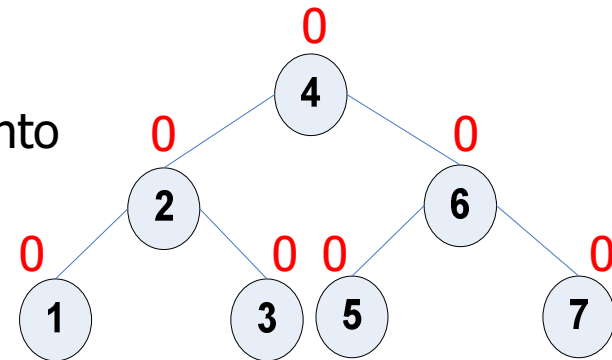
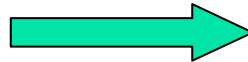


Operação: Inserção

Inserção: 4, 6, 1, 7, 5, 3 e 2.

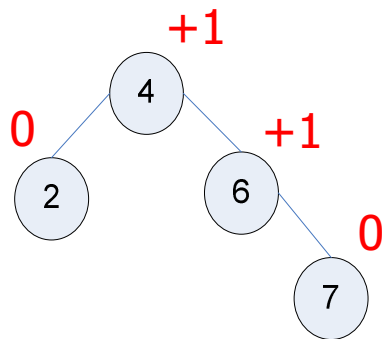


Op. de balanceamento

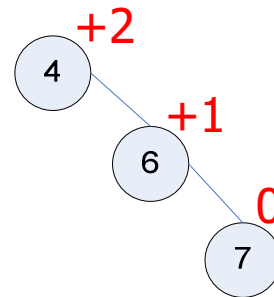
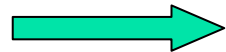


Operação: Remoção

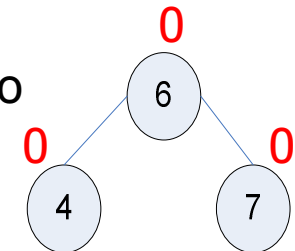
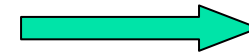
Inserção: 4, 6, 2 e 7.

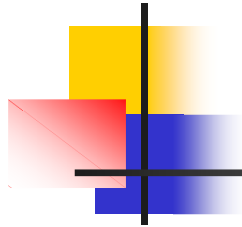


Remover nó 2



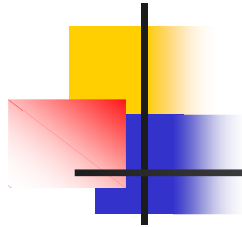
Op. de balanceamento





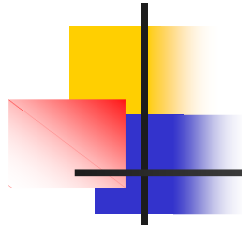
Roteiro

- Contextualização
- Árvores Balanceadas (AVL)
- **Operações de Balanceamento**



Operações de Inserção e Remoção

- A inserção ou remoção de um nó em uma árvore AVL **pode** ou **não** provocar seu desbalanceamento.
- Se a árvore AVL ficar desbalanceada, a restauração do seu balanceamento é realizado através de **ROTAÇÕES**.



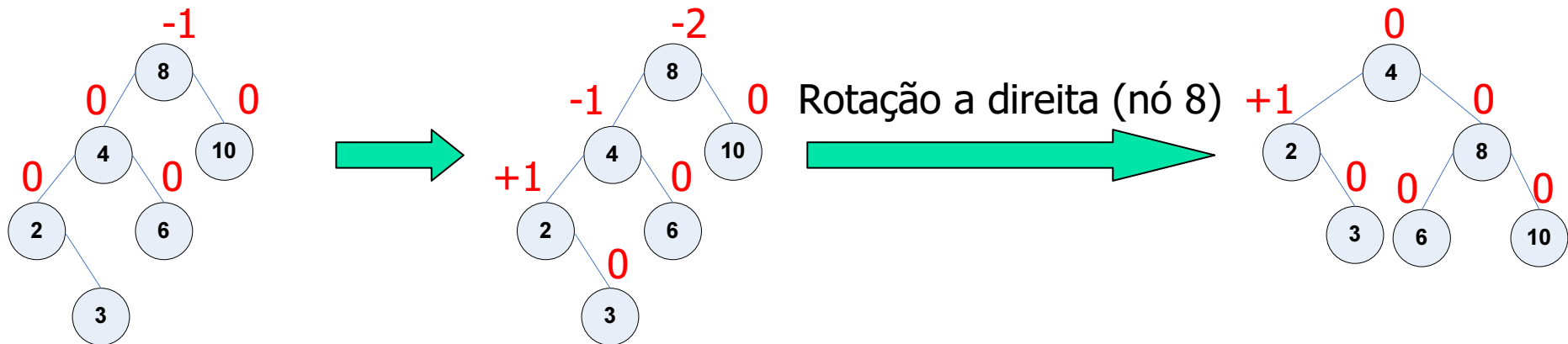
Tipos de Rotações

- Rotação Simples:
 - Rotação a Esquerda
 - Rotação a Direita

- Rotação Dupla:
 - Rotação a Esquerda
 - Rotação a Direita

Exemplos de Rotação Simples

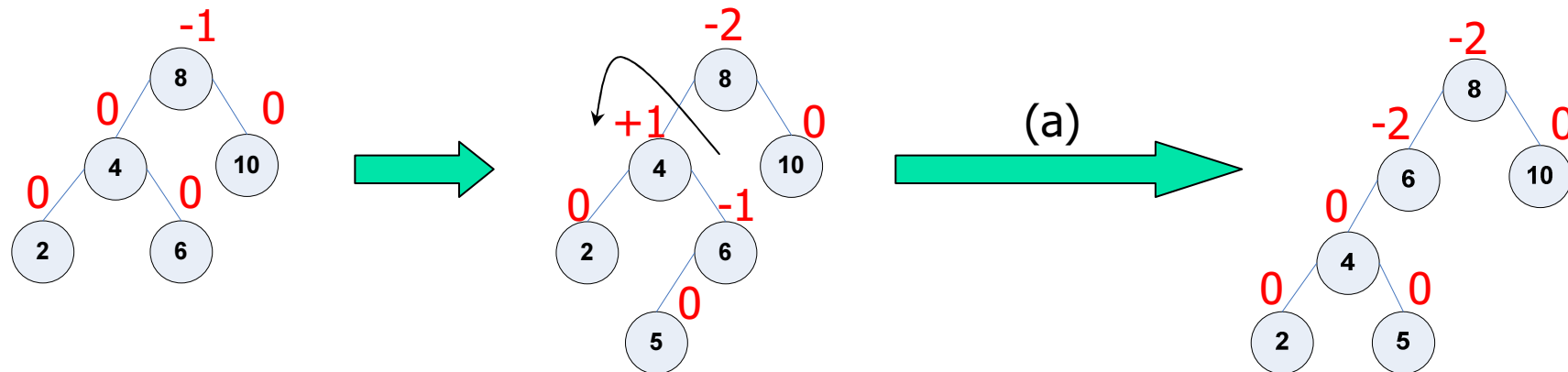
- Suponha que nós queiramos inserir o nó 3 na árvore inicial abaixo



A inserção do **nó 3** produziu um desbalanço no **nó 8** verificado pelo $FB = -2$ neste nó. Neste caso, como os sinais dos FB **são os mesmos** (**nó 8** com $FB = -2$ e **nó 4** com $FB = -1$) significa que precisamos fazer apenas uma **ROTAÇÃO SIMPLES**.

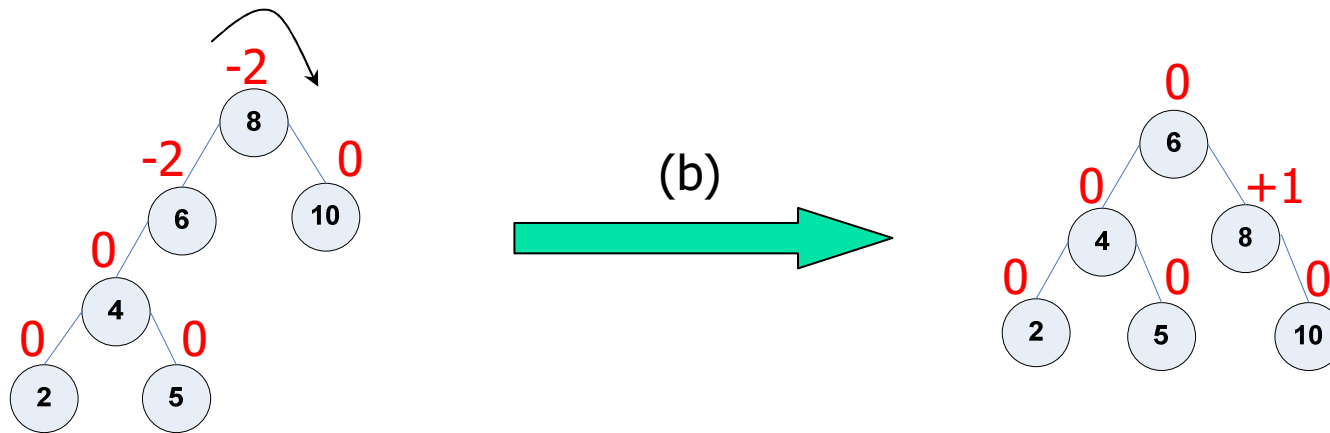
Exemplo de Rotação Dupla (1/2)

- Suponha que queiramos inserir o nó 5 na árvore abaixo



Observe que o **nó 8** tem $FB = -2$ e tem um filho com $FB = +1$ (**sinais opostos**). Neste caso, o balanceamento é alcançado com **duas rotações**. Primeiro: (a) rotação simples sobre o **nó 4** (com $FB = +1$) para a esquerda.

Exemplo de Rotação Dupla (2/2)



Logo após da rotação a esquerda: (b) rotaciona-se o **nó 8** (FB = -2) na direção oposta (direita neste caso).

Pseudo-Código: Rotações Simples

■ Rotação Simples a Esquerda

p aponta para o nó desbalanceado

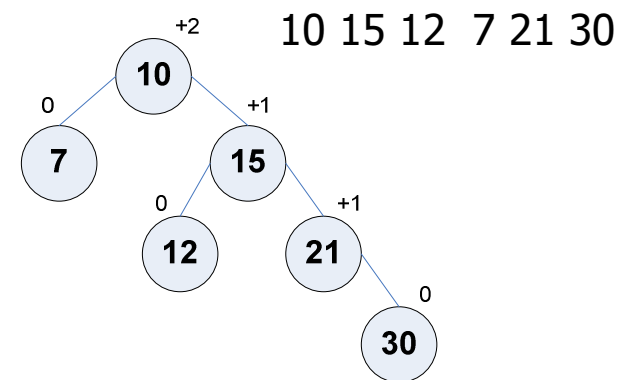
q = **right**(p);

hold = **left**(q);

left(q) = p;

right(p) = hold;

p = q;



■ Rotação Simples a Direita

p aponta para o nó desbalanceado

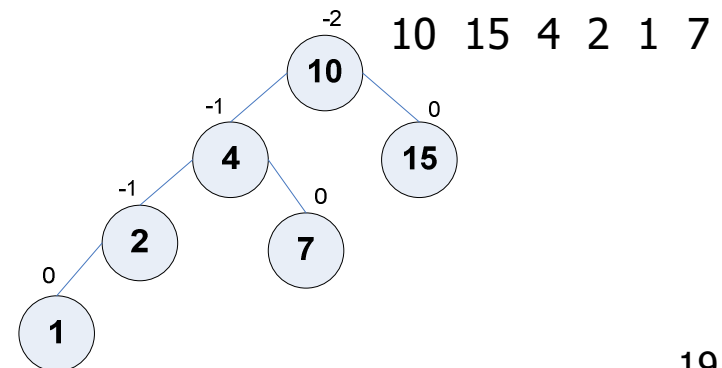
q = **left**(p);

hold = **right**(q);

right(q) = p;

left(p) = hold;

p = q;

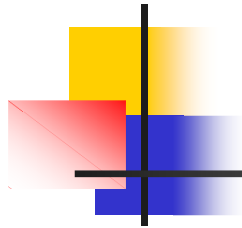




Pseudo-Código: Busca e Inserção

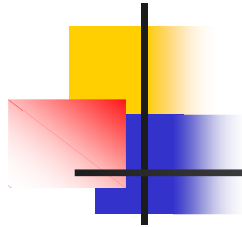
- **Busca e Inserção**

- Procurar **pseudo-código** no livro do Tenenbaum "Estrutura de Dados Usando C". pags: 531, 532, 533 e 534.



Conclusões

- Balanceamento de árvores **busca minimizar** o número **médio de comparações** necessárias para localizar qualquer dado.
- Operações de **inserção** e **remoção** de nós tendem a tornar as árvores **desbalanceadas**.
- Há um **custo extra de processamento**.
- Compensado quando os dados armazenados precisam ser recuperados **muitas vezes**.



AVL Tree Applet

- <http://webpages.ull.es/users/jriera/Docencia/AVL/AVL%20tree%20applet.htm>