

Estrutura de Dados

Prof. Dr. Gedson Faria

Prof.^a Dr.^a Graziela Santos de Araújo

Prof. Dr. Jonathan de Andrade Silva



Módulo 3 - Árvores Balanceadas

Unidade 1 - Conceitos: árvore AVL e balanceamentos por rotação



- Nós vimos que com as árvores binárias podemos reduzir o custo das operações de inserção, busca e remoção de $O(N)$ para $O(\log_2 N)$, mas desde que a árvore não seja degenerada;
 - Depende da ordem de inserção dos valores.
- Como evitar que as árvores binárias se tornem árvores degeneradas para garantirmos o custo $O(\log_2 N)$ independente da ordem de inserção dos valores?
 - Balanceamento da árvore binária.

Árvores Binárias Balanceadas

- Existem alguns tipos de árvore binárias que realizam esse processo de “balanceamento” para garantir a árvore binária quase-completa, por exemplo:
 - Árvores AVL (1962);
 - Árvores Rubro-Negras (1978);
 - Árvores *Splay* (1985).

AVL (Adelson-Velsky e Landis)

- Criada em 1962 por Georgy Adelson-Velsky e Evgenii Landis, a AVL é a mais antiga estrutura de dados para permitir o balanceamento das árvores binárias;
- Possui estratégias de balanceamento para permitir equilibrar as alturas das sub-árvores esquerda e direita de qualquer nó;
- O balanceamento é realizado sempre após as operações de inserção e remoção da árvore.

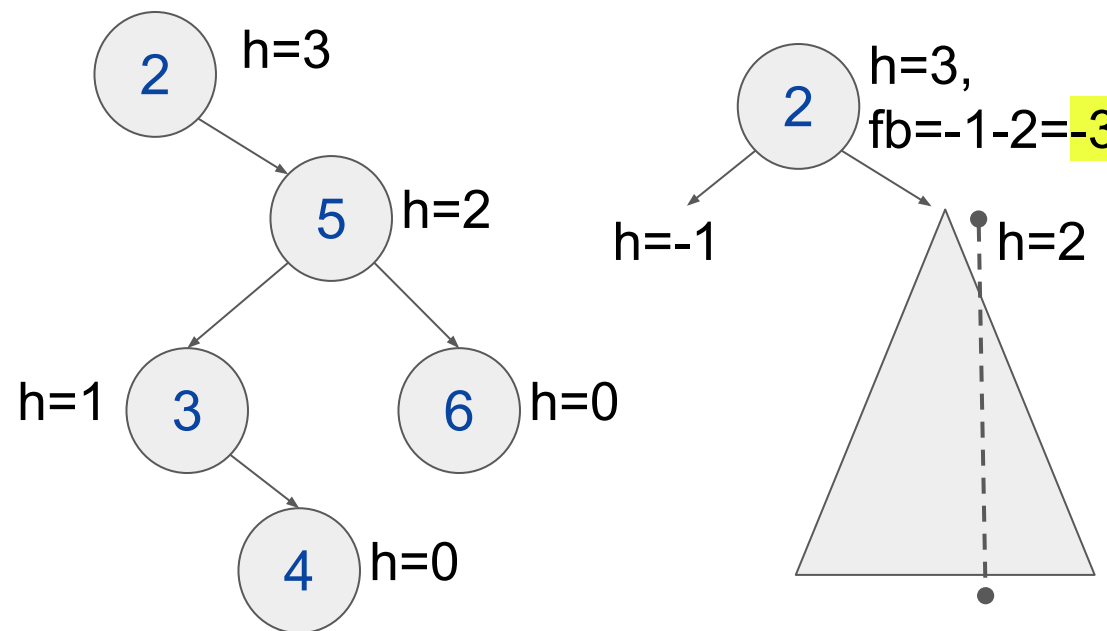
AVL (Adelson-Velsky e Landis)

- A estratégia de balanceamento envolve operações de **rotação** em qualquer nó da árvore;
- A ideia é rotacionar a árvore para esquerda ou direita para distribuir os nós de uma subárvore para outra.
- Na AVL uma árvore é dita balanceada quando:
 - A diferença de altura das subárvores esquerda e direita de um nó é de no máximo 1 unidade (± 1).

AVL - Balanceamento

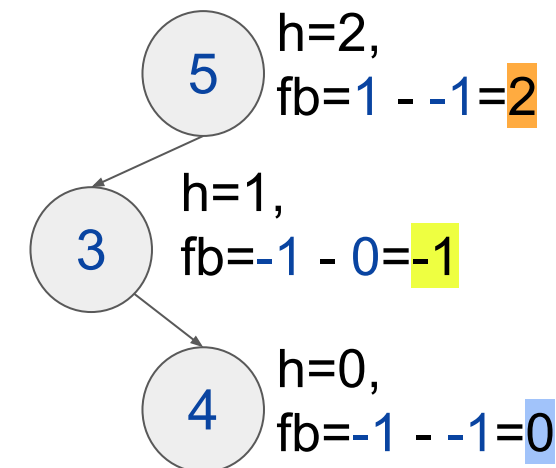
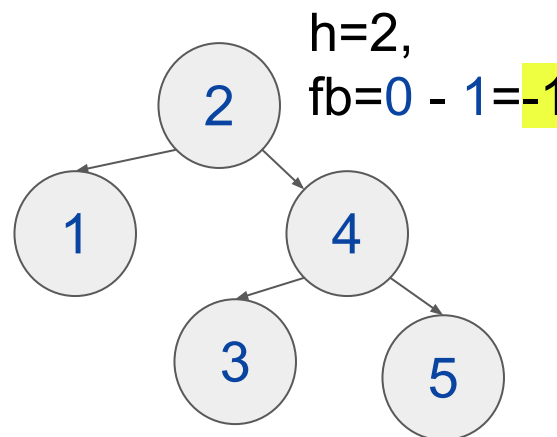
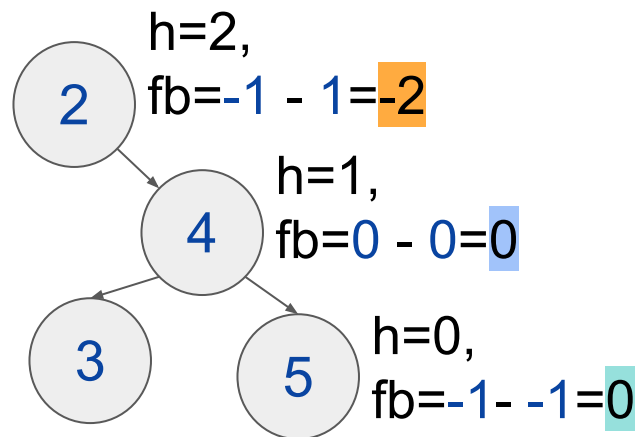
- A altura de um nó é o número de arestas no maior caminho até um nó folha;
- O fator de balanceamento (fb) é a diferença entre as alturas das subárvores esquerda e direita de um nó:

$$\text{fb}(\text{no}) = \text{altura}(\text{no.fe}) - \text{altura}(\text{no.fd})$$



AVL - Exemplos

- Calcular o fator de balanceamento dos nós e verificar se é AVL, (fb no máximo em 1 unidade).

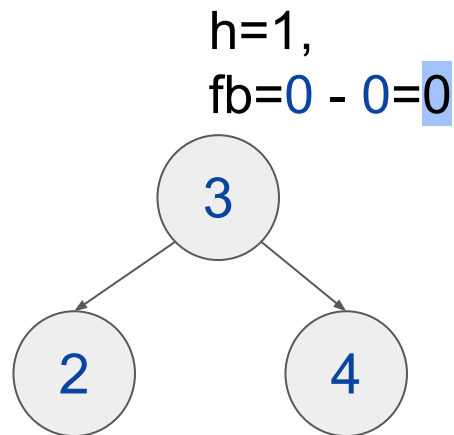
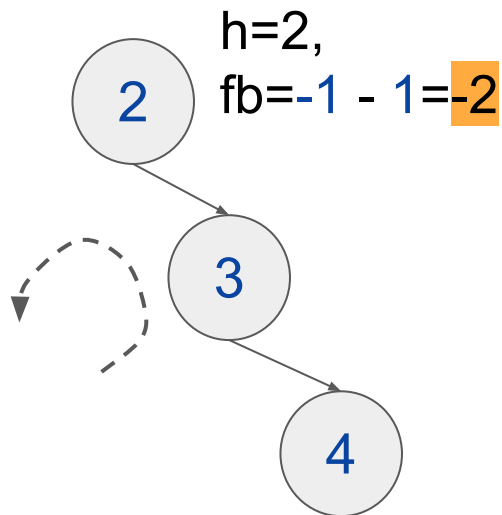


AVL - Balanceamento

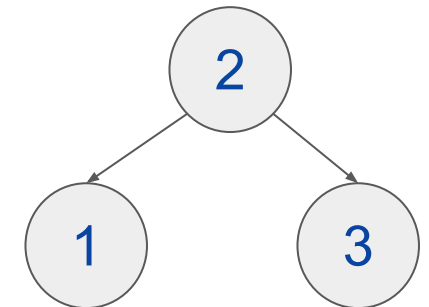
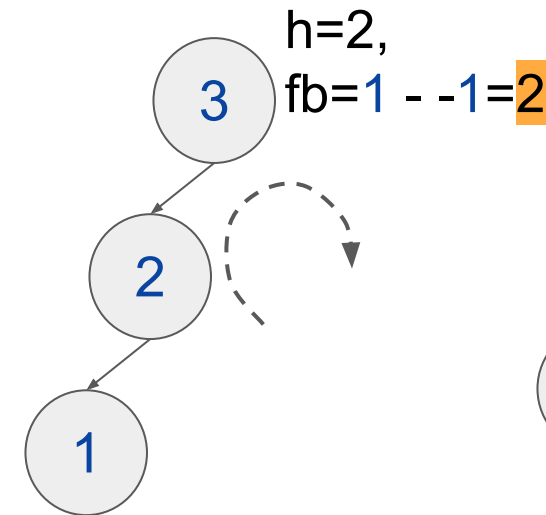
- Para realizar o balanceamento em um nó com fb com 2 unidades, temos 2 tipos de operação de rotação:
 - **Rotação Simples**: rotacionar para esquerda (RE) ou direita (RD) os nós da árvore.
 - **Rotação Dupla**: realizar duas rotações consecutivas e alternadas, rotação simples à esquerda seguida da direita (RED) ou direita seguida da esquerda (RDE).

Rotação Simples

- Rotação Simples à Esquerda (RE)



- Rotação Simples à Direita (RD)

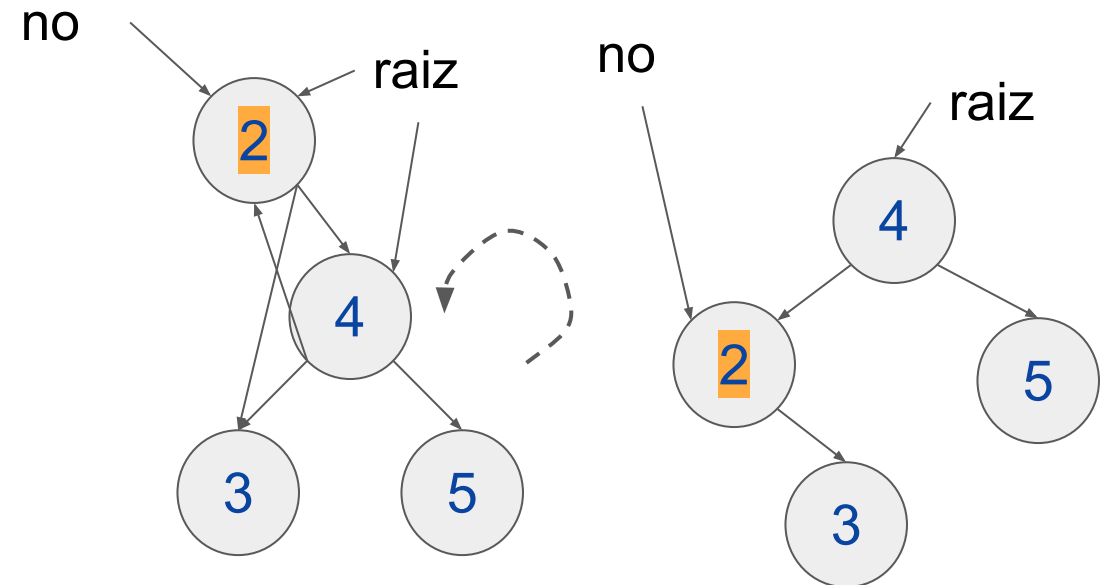


Rotação Simples

- Algoritmo RE(raiz):

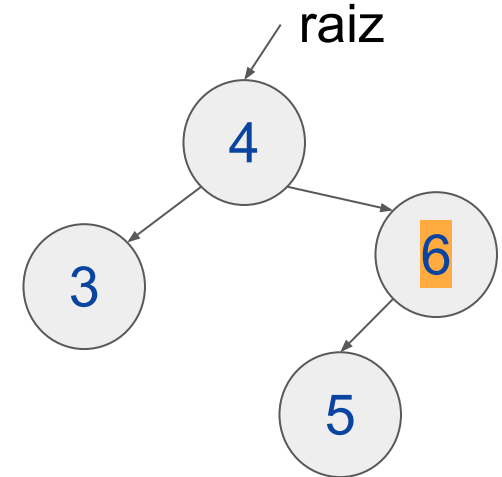
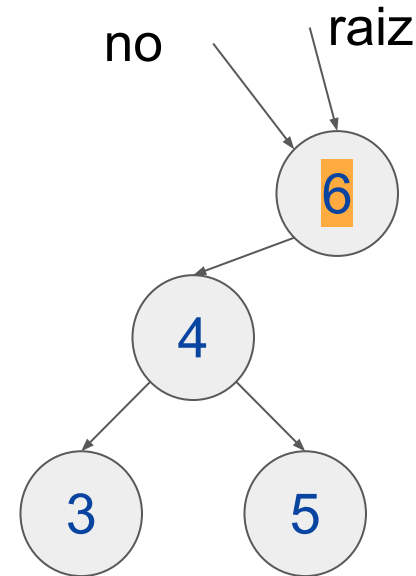
- ○ no = raiz
- ○ raiz = raiz.fd
- ○ no.fd = raiz.fe
- ○ raiz.fe = no
- retorna raiz

[Código](#)



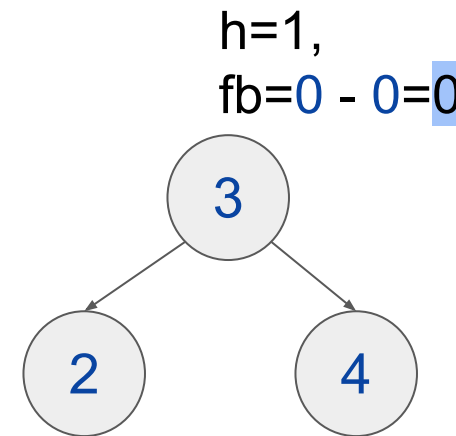
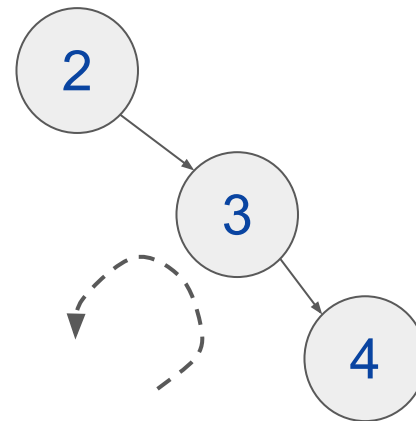
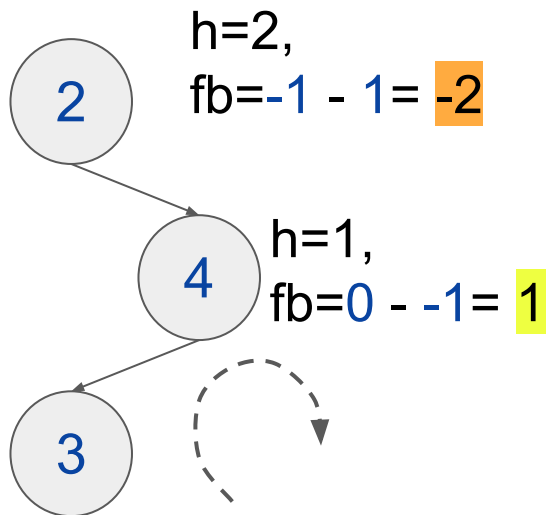
Rotação Simples

- Algoritmo RD(raiz):
 - $no = raiz$
 - $raiz = raiz.fe$
 - $no.fe = raiz.fd$
 - $raiz.fd = no$
 - retorna raiz



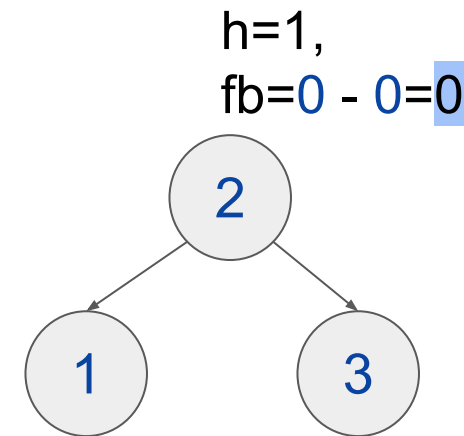
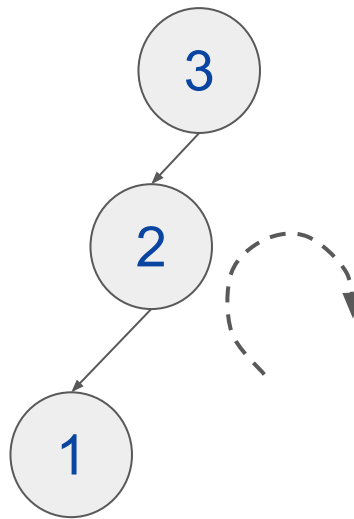
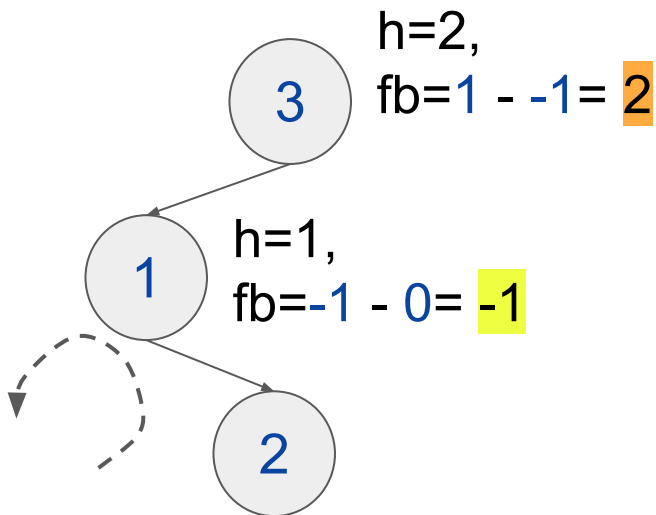
Rotação Dupla

- Rotação Dupla à Esquerda: Direita-Esquerda (RDE)



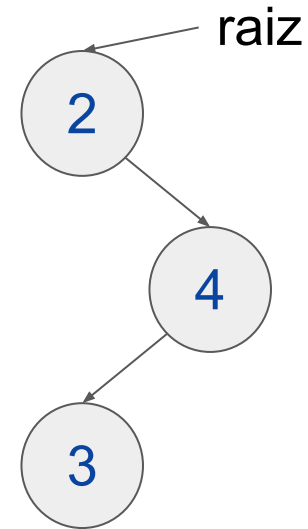
Rotação Dupla

- Rotação Dupla à Direita: Esquerda-Direita (RED)



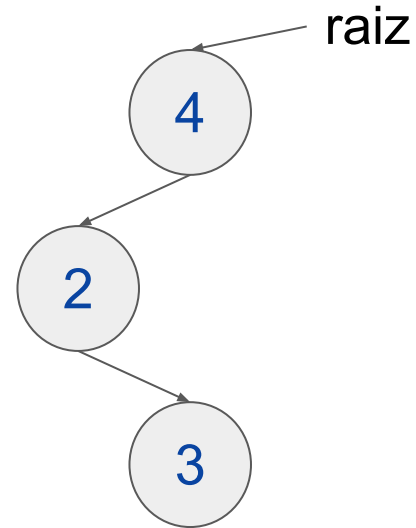
Rotação Dupla

- Algoritmo RDE(raiz):
 - raiz.fd = RD(raiz.fd)
 - raiz = RE(raiz)



Rotação Dupla

- Algoritmo RED(raiz):
 - $\text{raiz.fe} = \text{RE}(\text{raiz.fe})$
 - $\text{raiz} = \text{RD}(\text{raiz})$

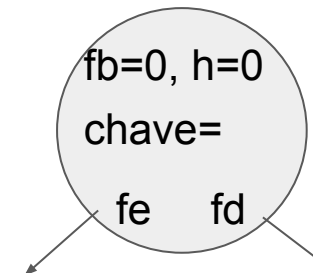


AVL - Inserção e Remoção

- O procedimento de inserção/remoção é realizado como na árvore binária.
 - Porém, atualiza-se o fator de balanceamento (fb) dos nós.
- Caso o fb exceda em uma unidade ($fb \pm 1$) aplicar as rotações adequadas para o devido balanceamento da árvore.

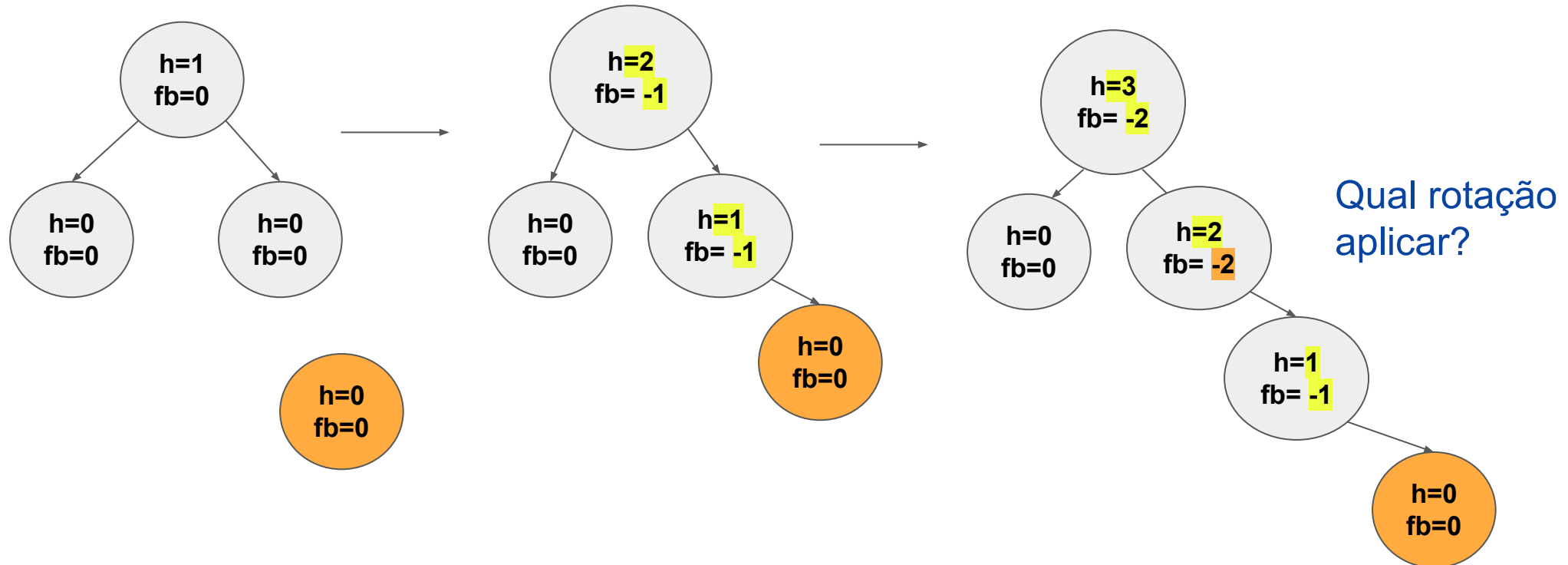
AVL - Componentes do Nó

- Podemos então definir que o nó seja descrito por:
 - **chave**: valor a ser inserido;
 - **fb**: fator de balanceamento;
 - **h**: altura do nó;
 - **fe**: filho esquerdo;
 - **fd**: filho direito.



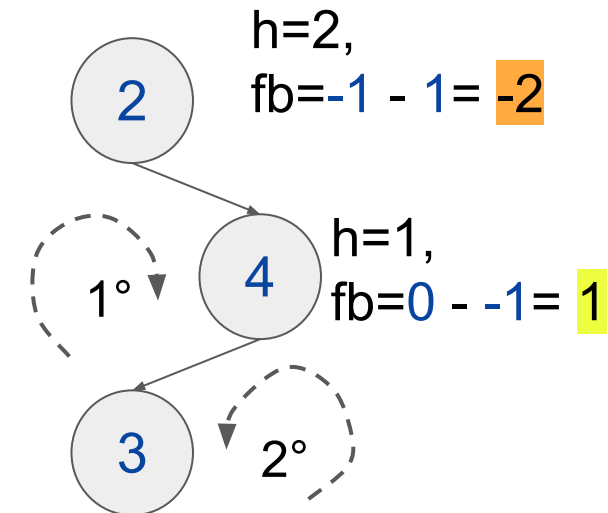
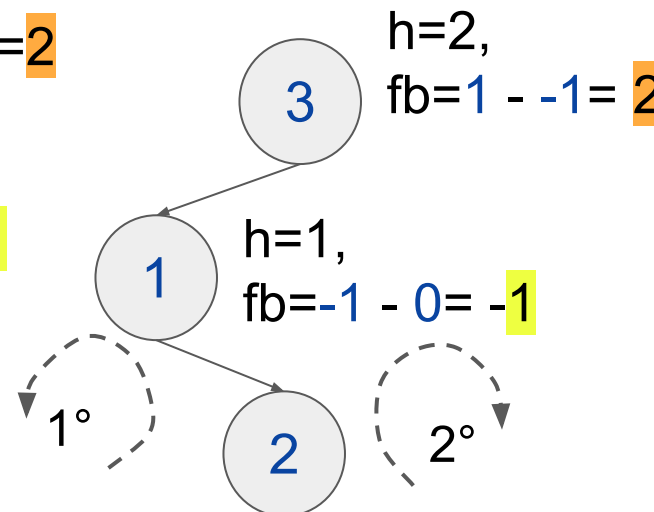
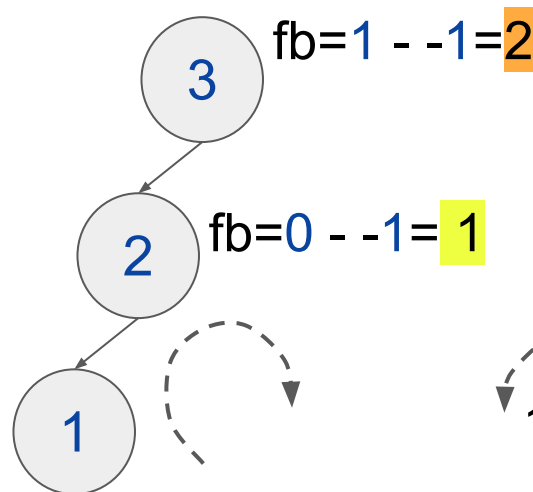
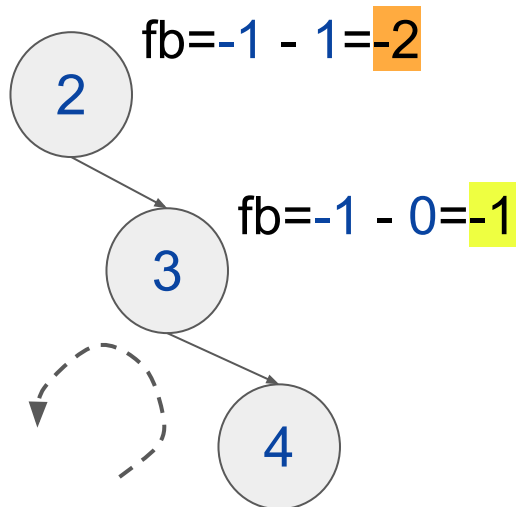
AVL - Inserção

- Vamos considerar a inserção de 2 nós na árvore abaixo.



AVL - Inserção

- Precisamos descobrir como saber qual situação aplicar as rotações;
- Temos que observar o fb dos nós.



AVL - Inserção

- Precisamos descobrir como saber qual situação aplicar as rotações:

RE	RD	RED	RDE
$fb(no) = -2$	$fb(no) = 2$	$fb(no) = 2$	$fb(no) = -2$
$fb(no.fd) = -1$	$fb(no.fe) = 1$	$fb(no.fe) = -1$	$fb(no.fd) = 1$
mesmo sinal	mesmo sinal	sinais diferentes	sinais diferentes

AVL - Inserção

- Etapas:
 - Inserir o nó como na árvore binária;
 - Atualizar o **fb** dos nós do caminho da inserção;
 - Verificar quando houve desbalanceamento;
 - Aplicar o procedimento de rotação adequado.
- Exemplo no [VISUALGO](#).
 - Testar com os casos de rotação simples e duplo.
- Na próxima aula vamos trabalhar nos detalhes do algoritmo de inserção e remoção.

Referências

CORMEN, Thomas. **Algoritmos: teoria e prática**. Rio de Janeiro: GEN LTC, 2013. ISBN 9788595158092. Disponível na Biblioteca Digital da UFMS.

SZWARCFITER, Jayme Luiz; MARKENZON, Lilian. **Estruturas de dados e seus algoritmos**. 3. ed. Rio de Janeiro, RJ: LTC, 2010. ISBN 9788521629955. Disponível na Biblioteca Digital da UFMS.

Licenciamento



Respeitadas as formas de citação formal de autores de acordo com as normas da ABNT NBR 6023 (2018), a não ser que esteja indicado de outra forma, todo material desta apresentação está licenciado sob uma [Licença Creative Commons - Atribuição 4.0 Internacional](https://creativecommons.org/licenses/by/4.0/).