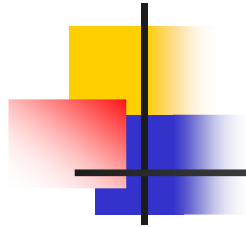




Algoritmos e Estruturas de Dados

Prof. Marcelo Zorzan
Profa. Melissa Zanatta



Aula de Hoje

- Árvore n-ária
- Árvore AVL



Implementação de uma Árvore n-ária

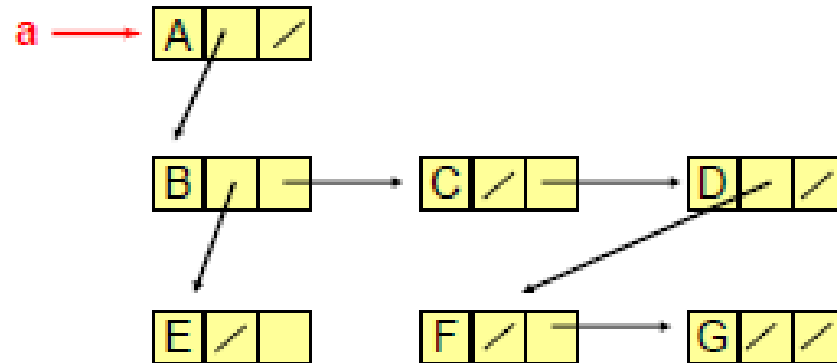
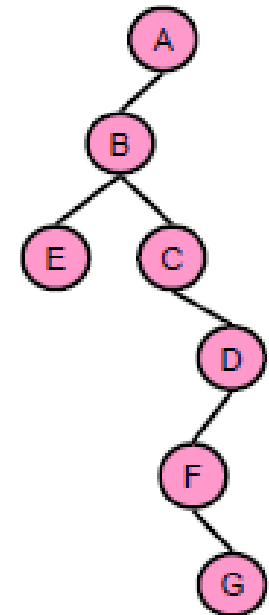
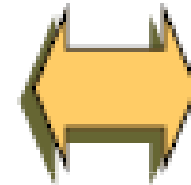
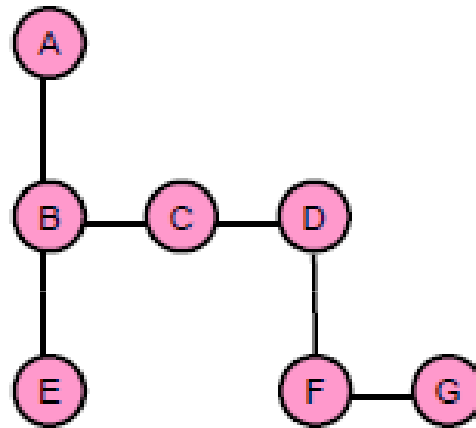
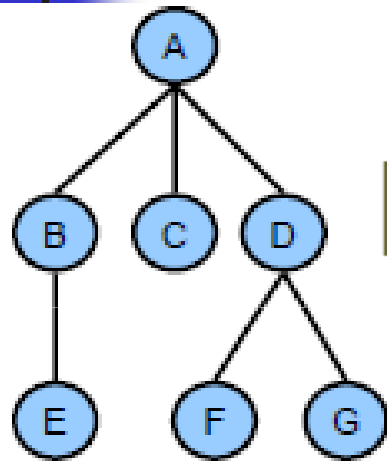
- Como seria a implementação de uma árvore n-ária?
- Nesse tipo de árvore cada nó pode conter um número diferente de sub-árvores.
- ✓ Solução 1: criar ponteiros de acordo com o grau da árvore, ou seja, se a árvore tiver grau 5, cinco ponteiros serão criados.
 - Desvantagem: Para os nós com grau menor que 5 acontecerá de alguns ponteiros nunca serem usados.
- ✓ Solução 2: transformar a árvore n-ária em árvore binária



Conversão de Árvore n-ária em Árvore binária

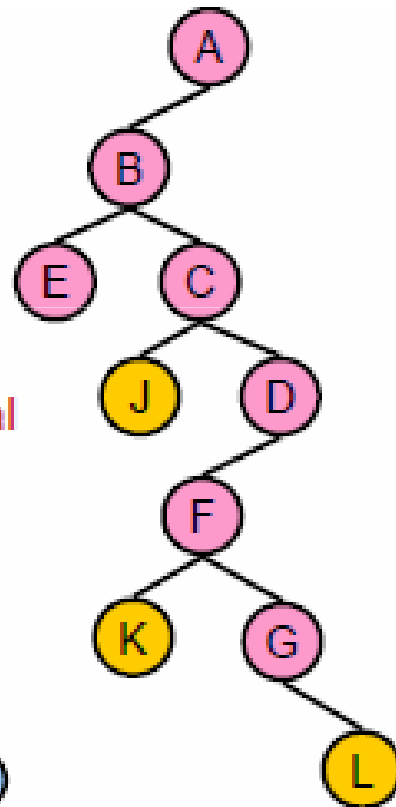
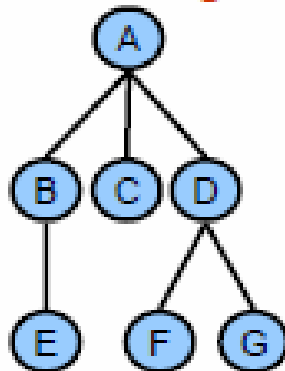
- Passos:
 - 1) O primeiro filho de um nó passa a ser seu filho à esquerda na árvore binária
 - 2) Os demais filhos de um nó passam a ser filhos à direita do seu irmão imediato à esquerda
 - 3) Executar o mesmo processo para cada nó da árvore
- Filho à esquerda = primeiro filho
- Filho à direita = irmão seguinte

Conversão de Árvore n-ária em Árvore binária

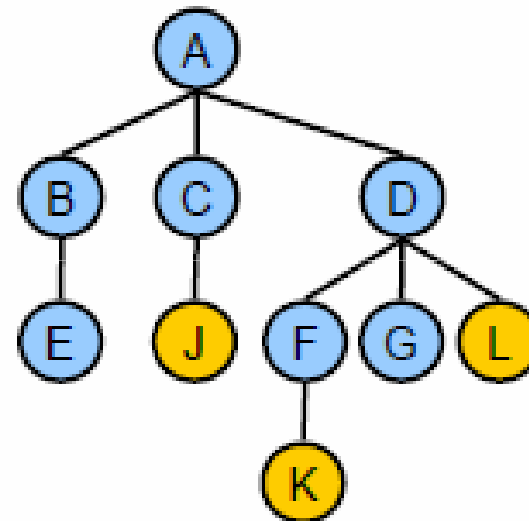


Reconstituição Árvore n-ária

Árvore original

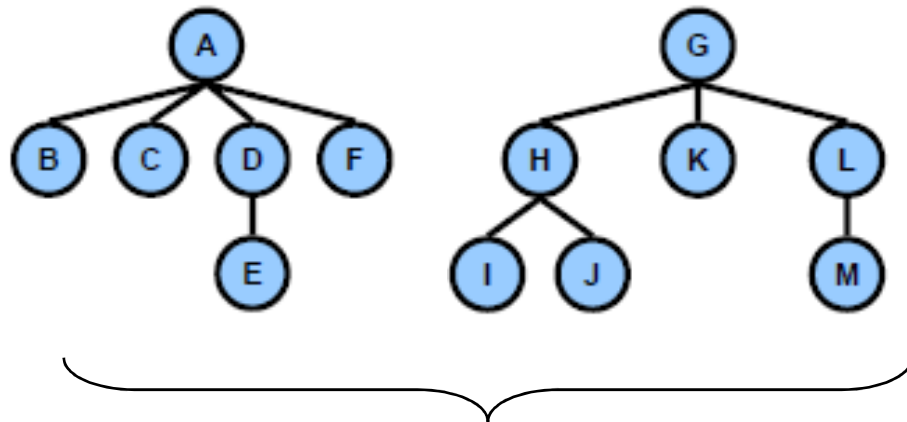


Árvore modificada

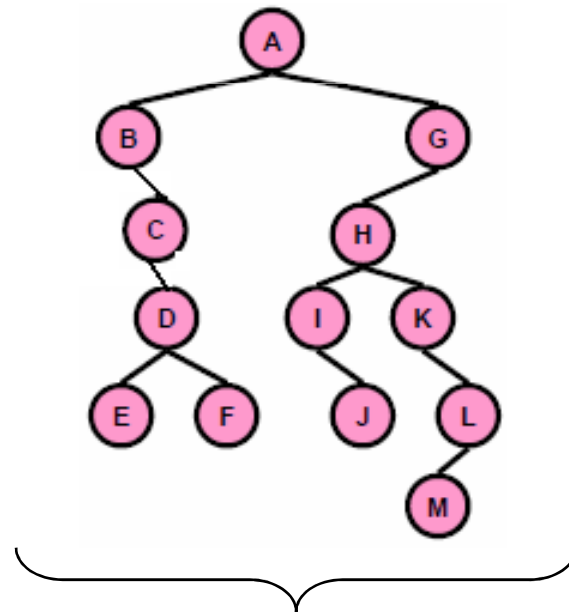


Transformação de Floresta em Árvore Binária

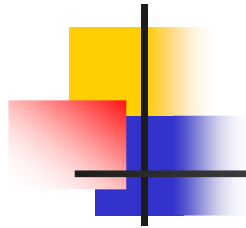
- Para converter uma floresta em árvore binária, basta considerar as raízes das árvores como nós irmãos e aplicar a conversão anterior



Floresta



Árvore Binária



Situação - Problema

Nilton aluno do curso de Sistemas de Informação sempre gostou de colecionar os gibis da turma da Mônica. Empolgado com a disciplina de Algoritmos e Estrutura de Dados ele decidiu implementar uma aplicação para cadastrar todos os seus gibis usando o TAD Árvore Binária de Pesquisa.

Como Nilton é muito cuidadoso ele sempre numerou os gibis em ordem crescente, ou seja, o primeiro gibi comprado era o número 1; o segundo, o número 2 e assim por diante. Como os gibis foram guardados nessa ordem, ele decidiu fazer o cadastro na aplicação considerando a numeração dos gibis em ordem decrescente.

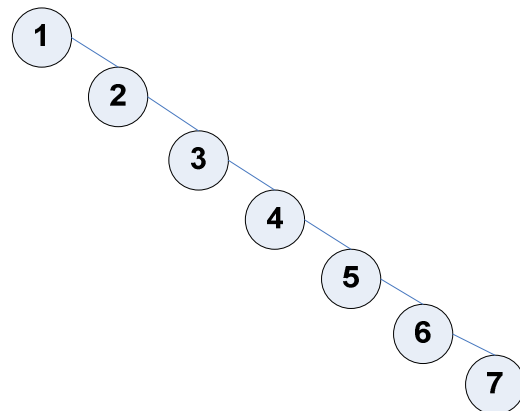
→ O que você acha dessa ordem de cadastro?



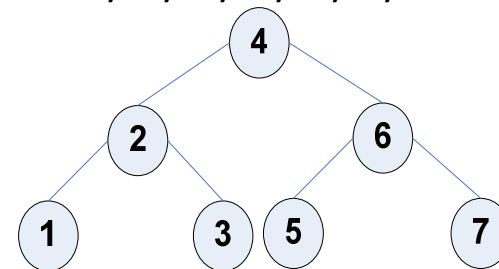
Árvore AVL - Introdução

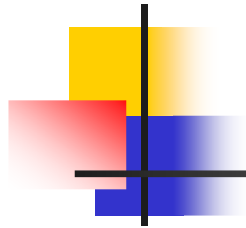
- As Árvores Binárias de Pesquisa estudadas até agora possuem uma **desvantagem** que pode afetar o tempo necessário para recuperar um item armazenado.
- A desvantagem é que o desempenho da Árvore Binária de Pesquisa **depende da ordem em que os elementos são inseridos**.

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



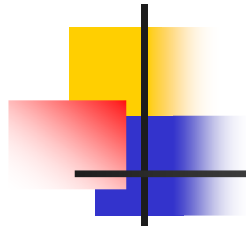
4, 6, 2, 5, 1, 7, 3





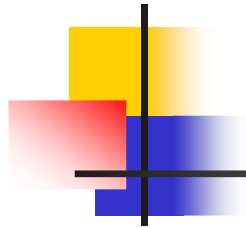
Árvore AVL - Introdução

- O ideal é que a árvore esteja **balanceada** para qualquer nó da árvore.
- Como saber se a árvore está balanceada ?
 - Para cada nó da árvore, a altura da sua subárvore esquerda deve ser **aproximadamente igual** à altura da sua subárvore direita.



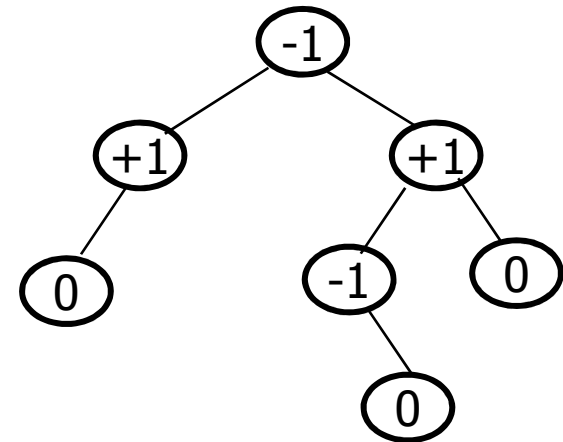
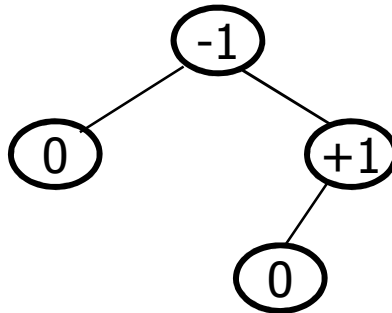
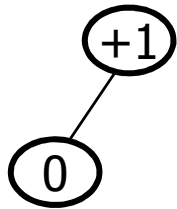
Árvore AVL - definição

- Também conhecida como “Árvore Balanceada pela Altura”
- O nome AVL vem das iniciais de seus inventores: Adelson-Velskii e Landis, que publicaram em 1962 o artigo: *“An algorithm for the organization of information”*.
- Uma árvore binária de pesquisa **T** é denominada **AVL** se:
 - Para todos nós de **T**, as alturas de suas duas subárvores diferem **no máximo de uma unidade**.



Árvore AVL

- Exemplos:

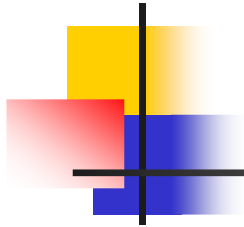


- O número dentro do nó indica o *fator de balanceamento*, que é a diferença entre a altura da subárvore esquerda e a altura da subárvore direita.



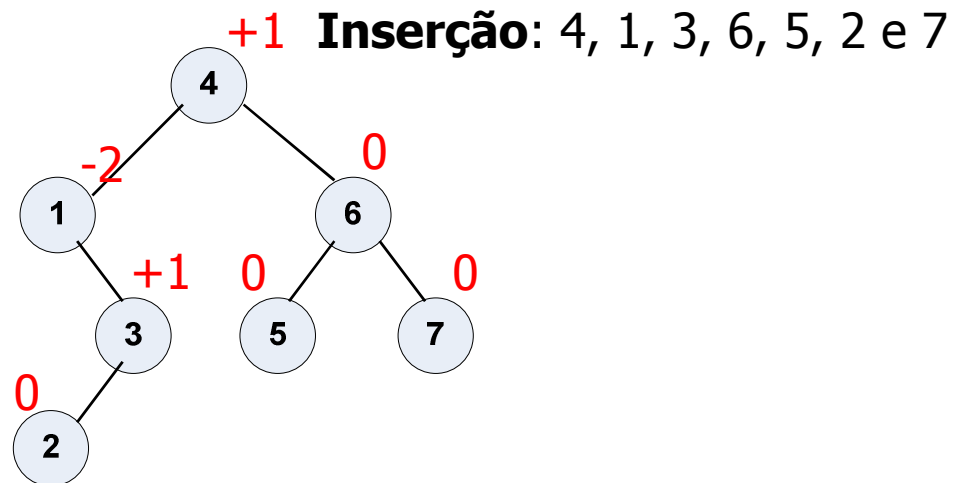
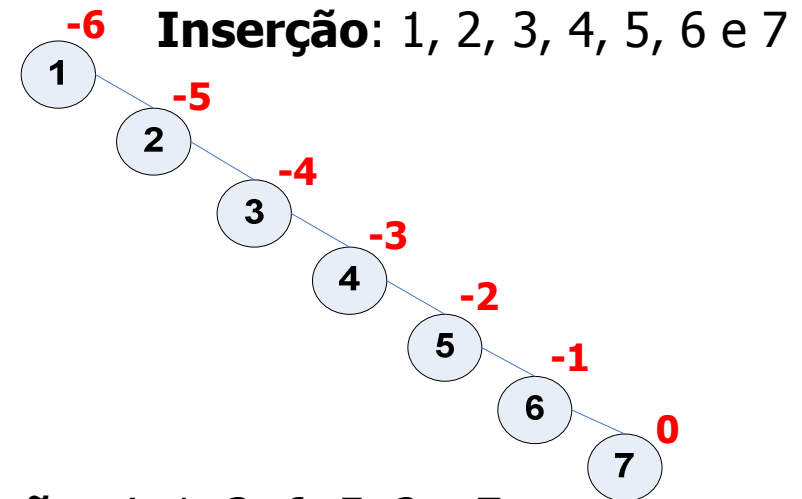
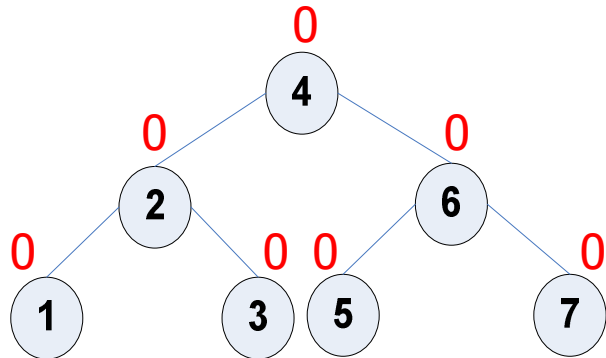
Fator de Balanceamento (FB)

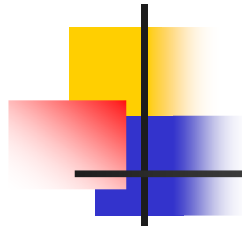
- O fator de balanceamento de um nó é a altura de sua subárvore esquerda menos a altura de sua subárvore direita.
- Um nó com fator de balanceamento 1, 0 ou -1 é considerado balanceado.
- Qualquer fator diferente de 1, 0, -1 caracteriza a árvore como não-AVL e indica necessidade de balanceamento por rotação simples ou rotação dupla.
- Este fator pode ser armazenado diretamente na estrutura ou calculado a partir da altura das subárvores.



Exemplos de cálculo do FB

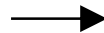
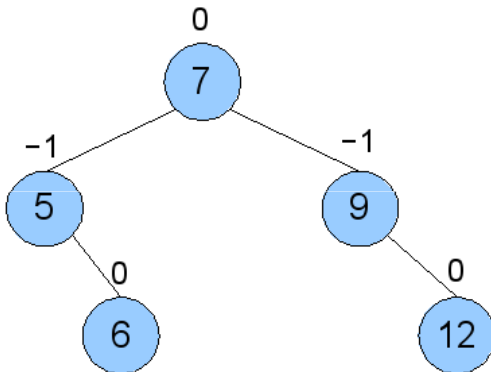
Inserção: 4, 2, 3, 6, 5, 1 e 7



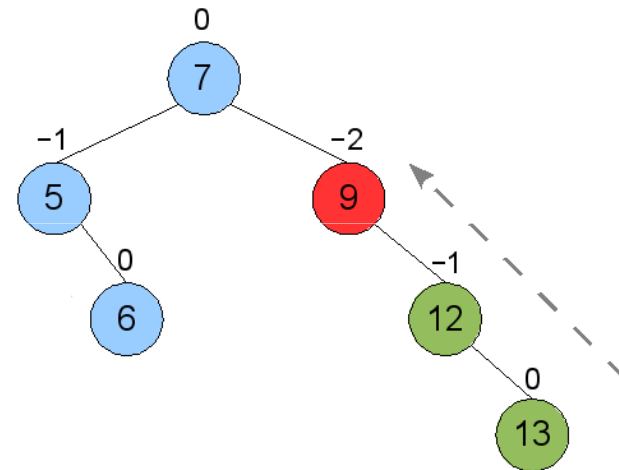


AVL – operação de inserção

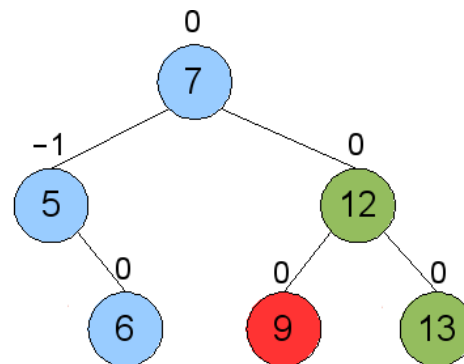
Inserção: 7, 5, 9, 6, 12

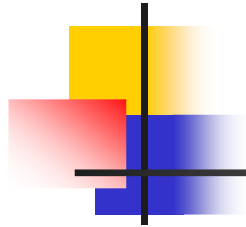


Inserção: 13



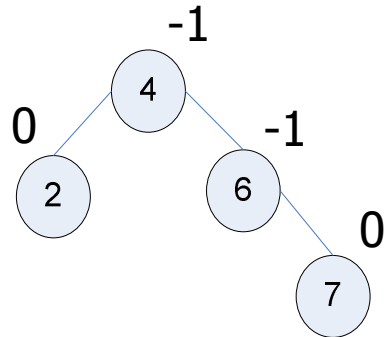
Operação de Balanceamento



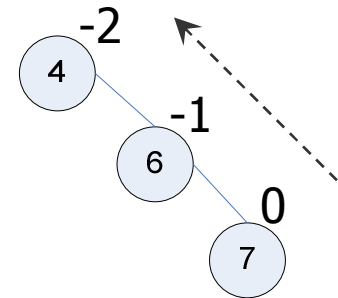


AVL – operação de remoção

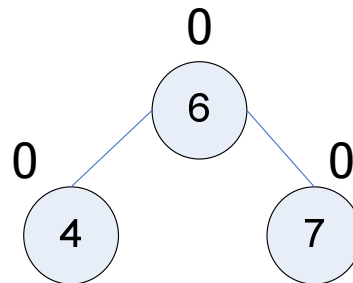
Inserção: 4, 6, 2 e 7.



Remoção: 2



Operação de Balanceamento





Árvore AVL

- Para manter uma árvore balanceada é necessário aplicar uma transformação na árvore que atendar às seguintes condições:
 - 1) o percurso em-ordem na árvore transformada seja igual ao da árvore original;
 - 2) a árvore transformada fique balanceada, ou seja, todos os nós com $FB = +1, 0$ ou -1



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** (simples ou dupla).



Árvore AVL - Inserção

- Inicialmente inserimos um novo nó na árvore.
- A inserção deste novo nó pode ou não violar a propriedade de balanceamento.
- Caso a inserção do novo nó não viole a propriedade de balanceamento podemos então continuar inserindo novos nós.
- Caso contrário, precisamos nos preocupar em restaurar o balanço da árvore através de rotações.



Árvore AVL - Inserção

- Se quisermos manter a árvore balanceada a cada inserção, devemos ter um algoritmo que ajuste os fatores de balanceamento.
- O algoritmo corrige a estrutura através de movimentação dos nós (rotações).

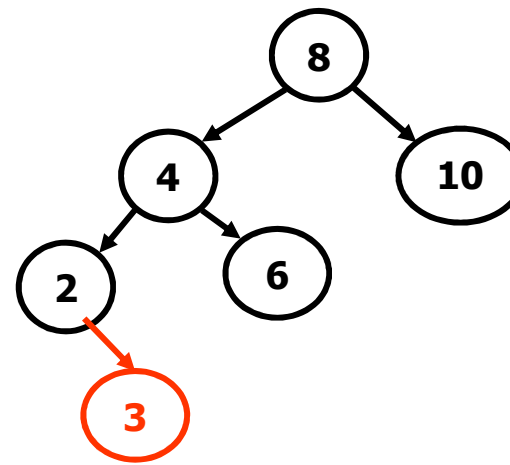
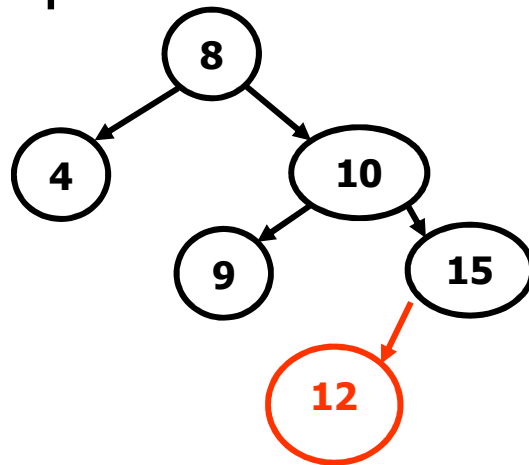


Árvore AVL - Inserção

- Os problemas de balanceamento das árvores AVL podem ser mapeados em dois casos:
 - **Caso 1:** o nó raiz de uma subárvore tem $FB=+2$ (ou -2) e tem um filho com $FB = +1$ (ou -1) o qual tem o mesmo sinal que o FB do nó pai.
 - **Caso 2:** o nó raiz de uma subárvore tem $FB=+2$ (ou -2) e tem uma um filho com $FB = -1$ (ou $+1$) o qual tem o sinal oposto ao FB do nó pai.

Árvore AVL - Caso 1

- Nó raiz da subárvore tem $FB=+2$ (ou -2) e tem filho com $FB=+1$ (ou -1) o qual tem o mesmo sinal que o FB do nó pai



- Solução: rotação simples sobre o nó de $FB=+2$ (ou -2).
 - Rotações são feitas à direita quando FB é positivo e à esquerda quando FB é negativo.

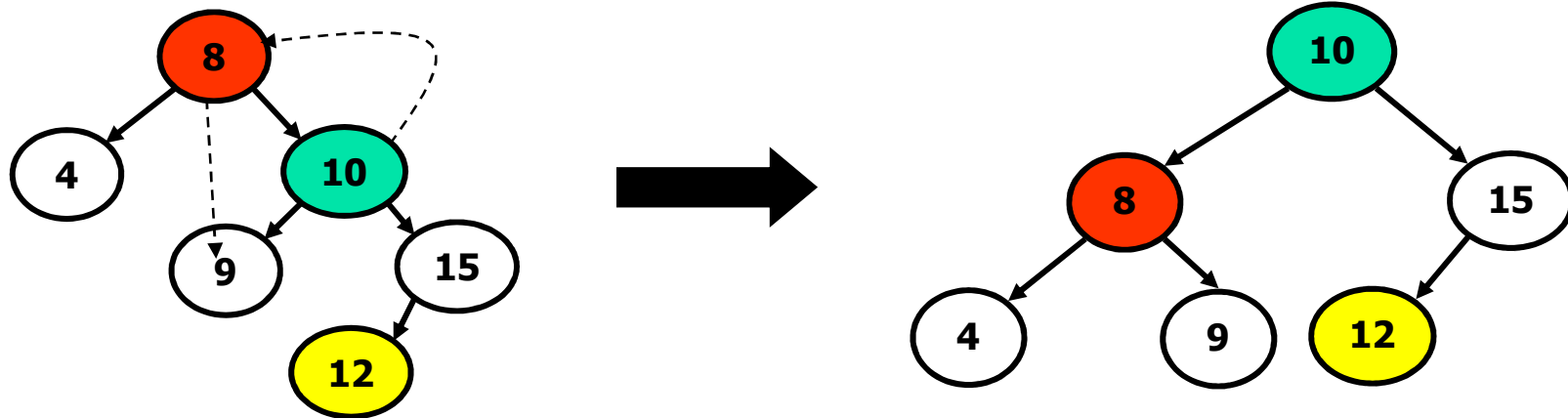
Árvore AVL – Caso 1

■ Exemplo 1:

$$FB(8) = 1 - 3 = -2$$

$$FB(10) = -1$$

→ Rotações são feitas à esquerda quando FB é negativo



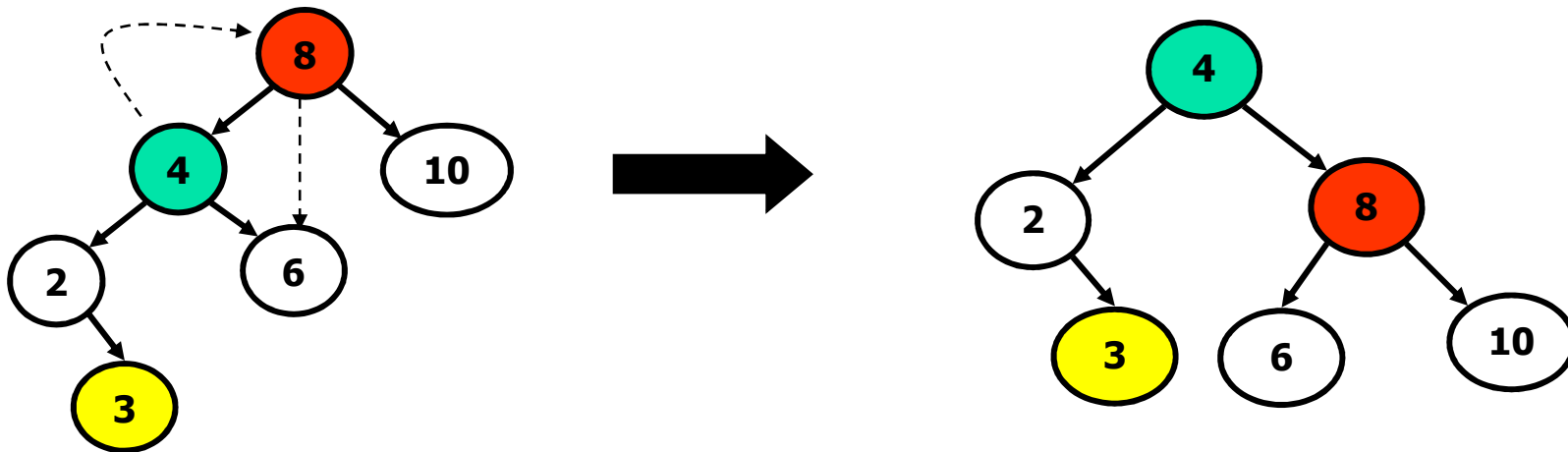
Árvore AVL – Caso 1

- **Exemplo 2:**

$$\text{FB}(8) = 3 - 1 = +2$$

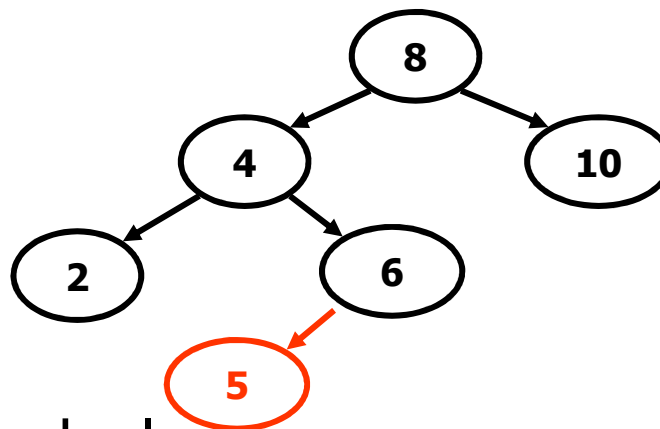
$$\text{FB}(4) = +1$$

→ Rotações são feitas à direita quando FB é positivo



Árvore AVL – Caso 2

- Nó raiz da subárvore tem $FB=+2$ (ou -2) e tem filho com $FB=-1$ (ou $+1$) o qual tem o sinal oposto ao do FB do nó pai



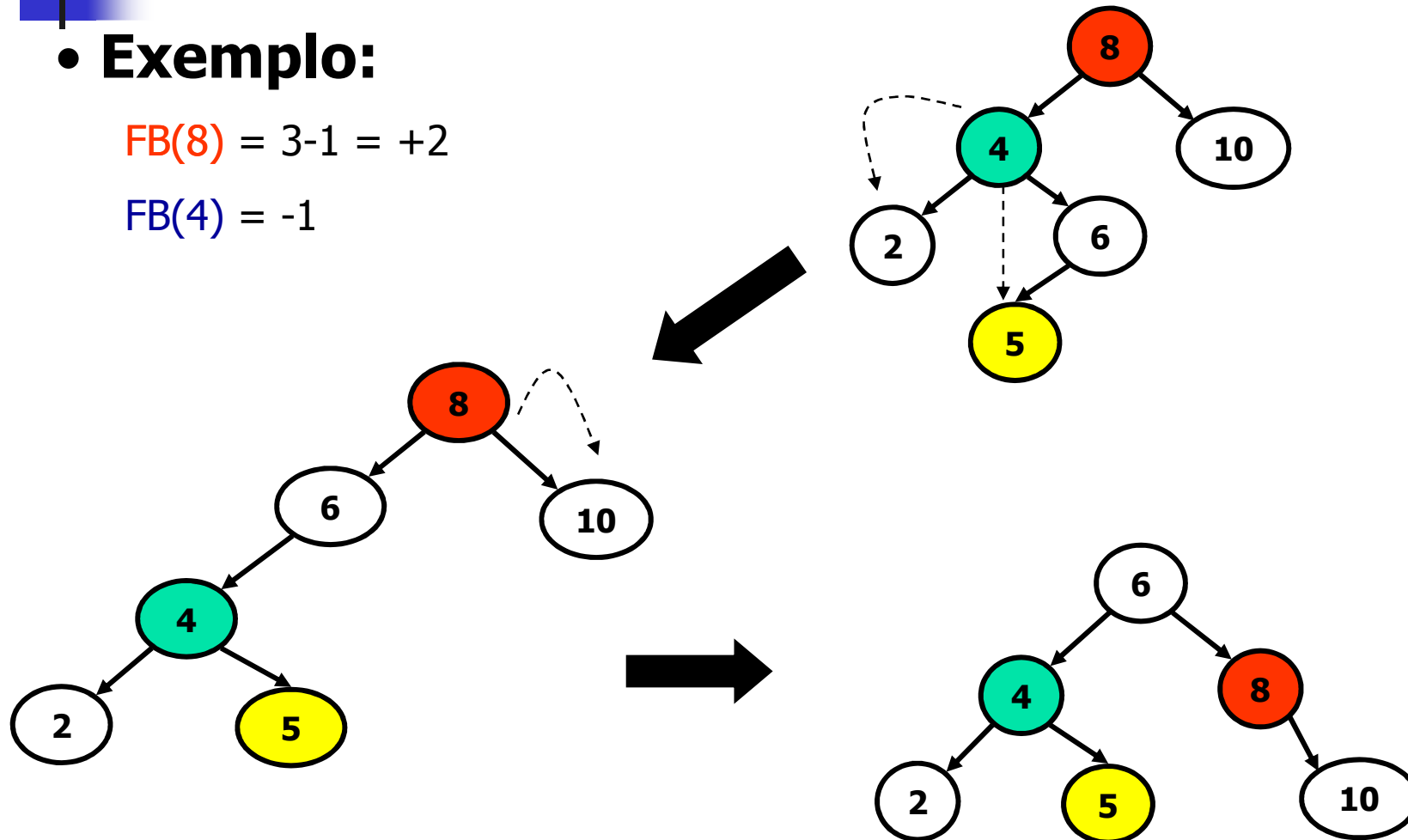
- Solução: rotação dupla
 - Primeiro a rotação sobre o nó com $FB=+1$ (ou -1) na direção apropriada (se FB negativo, para a esquerda; se positivo, para a direita)
 - Em seguida, a rotação sobre o nó com $FB=+2$ (ou -2) na direção oposta.

Árvore AVL – Caso 2

- **Exemplo:**

$$FB(8) = 3 - 1 = +2$$

$$FB(4) = -1$$





Árvore AVL – Resumo

- **Caso 1:**

- Rotacionar uma única vez o nó de $FB = -2$ ou $+2$:
 - se negativo: à esquerda;
 - se positivo: à direita



Árvore AVL – Resumo

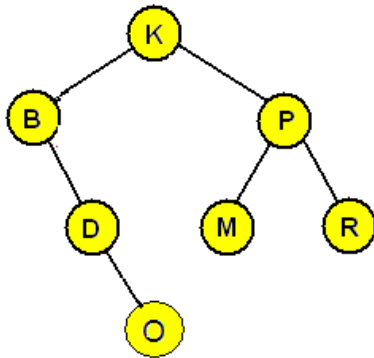
- **Caso 2:**

- Requer uma rotação dupla: ESQUERDA-DIREITA ou DIREITA-ESQUERDA:

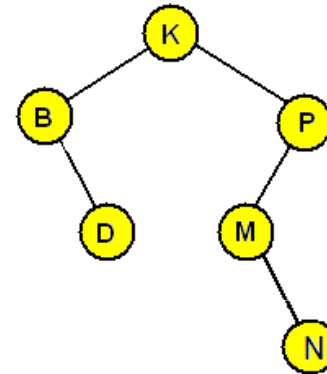
- 1) Rotacionar o nó com $FB = -1$ (ou 1) na direção apropriada, isto é, se FB negativo, para a esquerda; se positivo, para a direita.
- 2) Rotacionar o nó com $FB = -2$ (ou 2) na direção oposta.

Exercícios

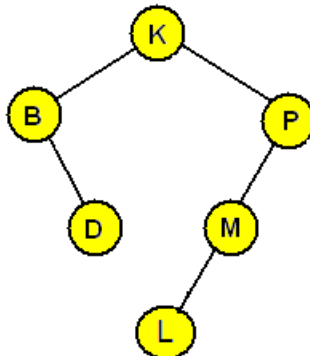
1)



2)



3)



4)

