

# Árvores

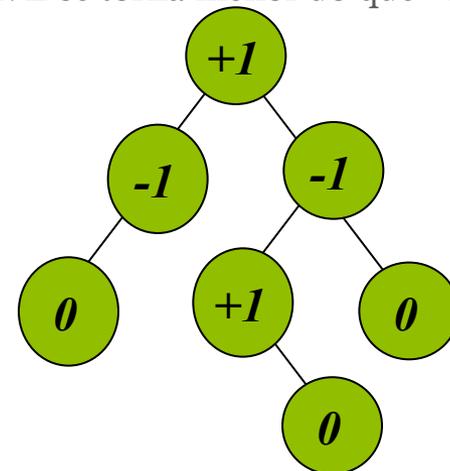
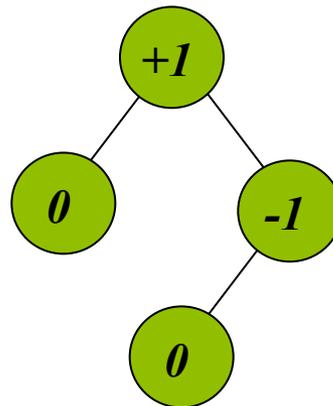
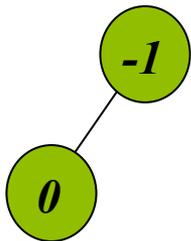


# Árvores AVL

- ◆ Os algoritmos vistos trabalham sobre a árvore toda.
  - ◆ Se houver a necessidade de manter o balanceamento a cada inserção ou remoção, então sua eficiência fica bastante prejudicada.
- ◆ O balanceamento da árvore pode ser realizado localmente se apenas uma porção da árvore for afetada por operações de inserção ou remoção.
- ◆ Uma alternativa para esse problema é fazer uso de algoritmos que trabalhem apenas em parte da árvore, a cada inserção ou remoção.
- ◆ Um método clássico para o **balanceamento local** de árvores foi proposto por Adelson-Velskii e Landis e é usado na árvore que recebe a nomenclatura : **AVL**.
- ◆ Uma árvore AVL, também conhecida como **árvore admissível** é uma árvore na qual as alturas das subárvores esquerda e direita de cada nó diferem no máximo por um.

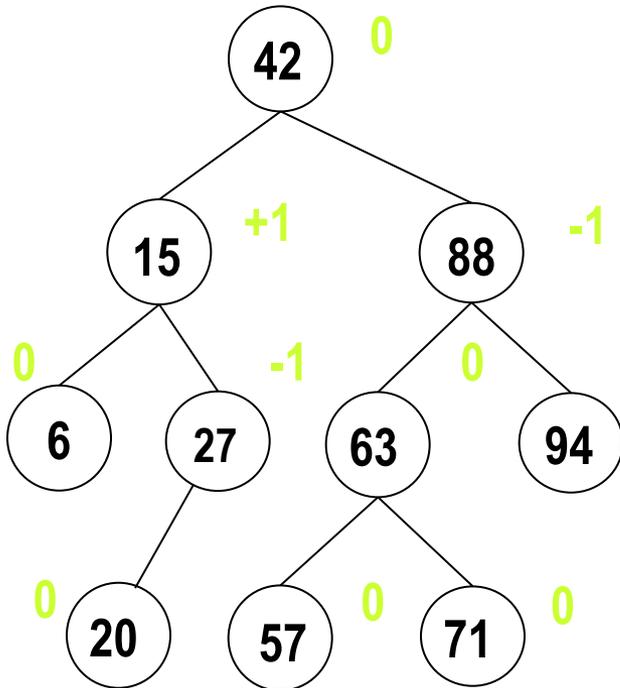
# Árvores AVL

- Os números nos nós indicam os fatores de balanceamento que são as diferenças entre as alturas das subárvores esquerda e direita.
- Numa árvore AVL todos os fatores de balanceamento devem ser **+1, 0 ou -1**.
- A definição de árvore AVL é semelhante à da árvore balanceada, mas as técnicas de balanceamento estão implícitas na definição.
- Se o fator de balanceamento de qualquer nó em uma árvore AVL se torna menor do que  $-1$  ou maior do que  $1$ , a árvore **tem que ser balanceada**.

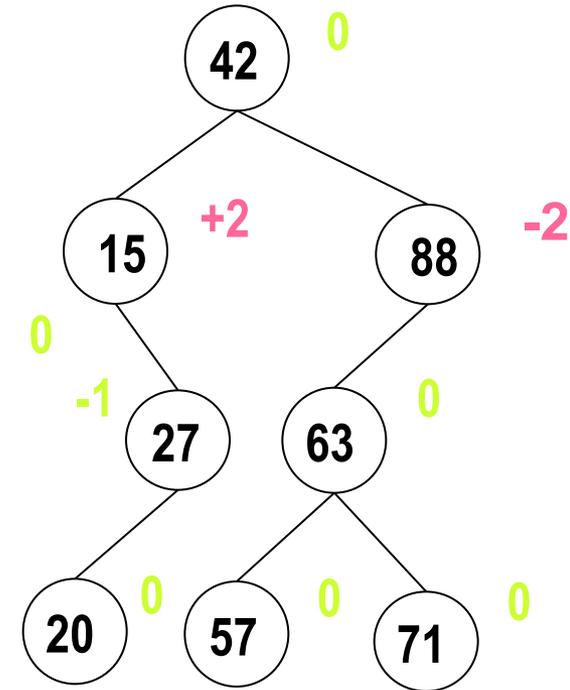
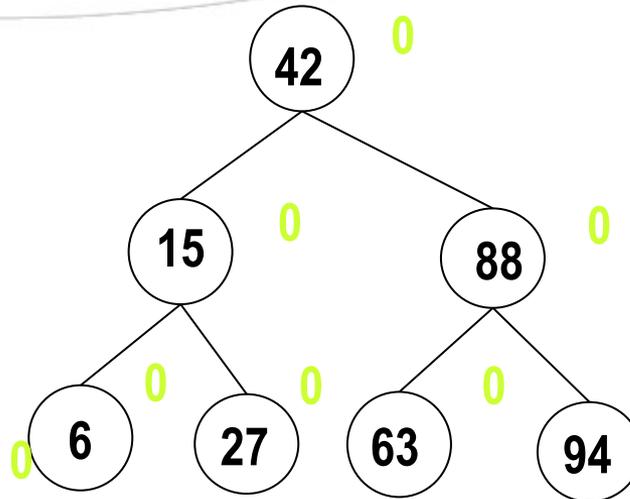


# Árvores AVL

Exemplos



Árvore AVL



Não é árvore AVL!

# Condição de balanceamento AVL

- ◆ Uma árvore vazia é balanceada AVL.
- ◆ Uma árvore não vazia  $T = \{r, TL, TR\}$  é balanceada AVL se tanto TL quanto TR forem balanceadas AVL e
- ◆  $|hL - hR| \leq 1$
- ◆ onde  $hL$  é a altura da árvore TL e  $hR$  é a altura da árvore TR.

# Balanceamento de Árvores AVL

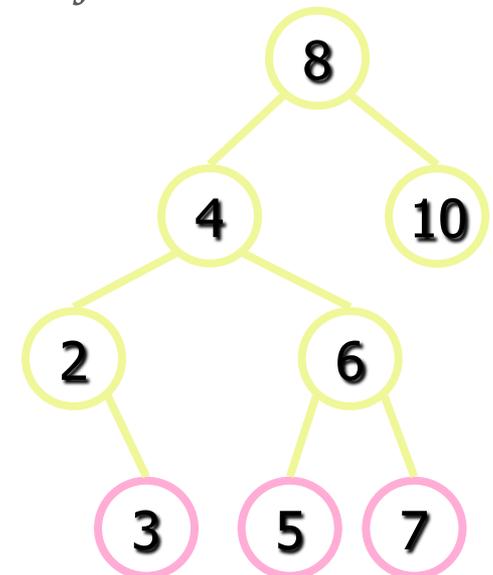
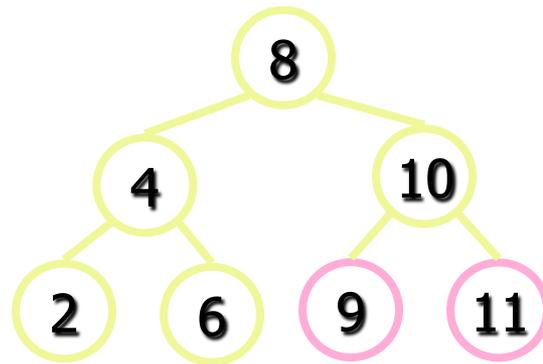
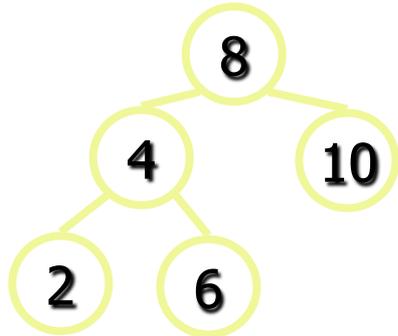
- ◆ Inserções e remoções podem destruir o equilíbrio de uma árvore.
- ◆ Após a operação ser efetuada é necessário verificar se houve quebra da propriedade de balanceamento da árvore.
  - ◆ A verificação do balanceamento da árvore é feita através do **controle das alturas** das sub-árvores esquerda e direita da raiz da árvore.
  - ◆ Se o módulo da diferença das alturas (altura da sub-árvore direita – altura da sub-árvore esquerda) for maior que 1,
    - ◆ houve a quebra do balanceamento, e
    - ◆ há a necessidade de re-estruturar a árvore para que ela volte a ser balanceada.

# Balanceamento de Árvores AVL

- ◆ Uma árvore AVL sempre irá procurar manter-se uma árvore binária quase completa, ou seja,
  - ◆ uma árvore tal que o número de passos (operações) necessários para se realizar qualquer algoritmo sobre a árvore seja **no máximo  $\log n$** .
- ◆ Dada uma **raiz  $r$**  com subárvores **L** (left) e **R** (right), e supondo que a inserção deve ser feita na sub-árvore da esquerda. Pode-se distinguir 3 casos:
  - ◆ **Se  $hL = hR$** , então L e R ficam com alturas diferentes mas continuam balanceadas.
  - ◆ **Se  $hL < hR$** , então L e R ficam com alturas iguais e balanceamento foi melhorado.
  - ◆ **Se  $hL > hR$** , então L fica ainda maior e balanceamento foi violado.

# Balanceamento de Árvores AVL

- ◆ Nós 9 ou 11 podem ser inseridos sem balanceamento. A subárvore com raiz 10 passa a ter duas subárvores e a árvore com raiz 8 vai ficar melhor balanceada!
- ◆ Inserção dos nós 3, 5 ou 7 requerem que a árvore seja rebalanceada!



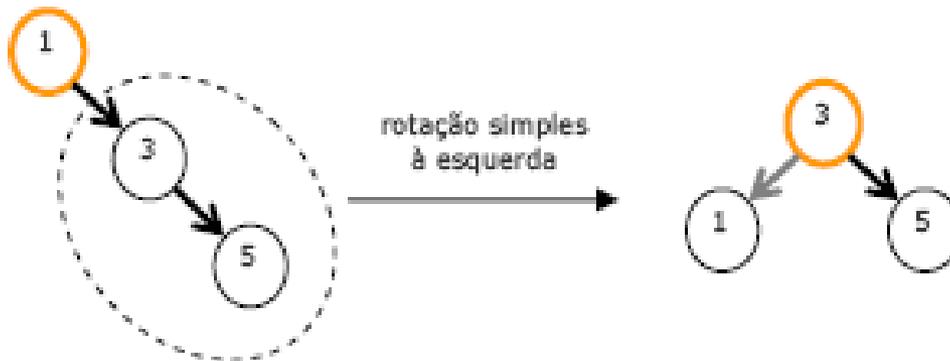
# Balanceamento de Árvores AVL

- ◆ A operação básica para balanceamento de uma árvore AVL é a **rotação**.
- ◆ Para o balanceamento da árvore pode-se realizar:
  - ◆ rotações simples ou
  - ◆ rotações duplas.
- ◆ Uma **rotação simples** ocorre quando um nó está desbalanceado e seu filho estiver no mesmo **sentido da inclinação**.
- ◆ Uma **rotação-dupla** ocorre quando um nó estiver desbalanceado e seu filho estiver inclinado no **sentido inverso ao pai**.

# Balanceamento de Árvores AVL

## ◆ Tipos de rotação

### ◆ 1º Caso: Rotação simples à esquerda

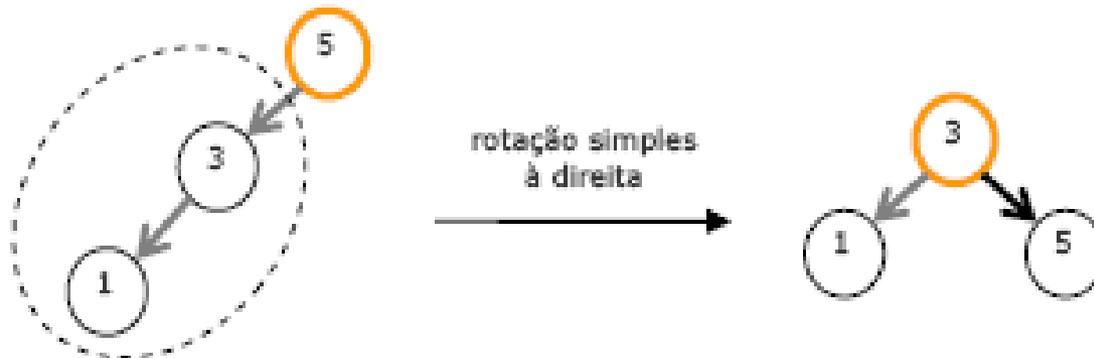


- O nó intermediário(3) é maior que seu pai(1), porém, é menor que seu filho(5).
- Logo o nó intermediário(3) deve ser escolhido para ser a raiz da árvore resultante.
- O nó corrente(1) tem que cair para a esquerda, ou seja, ser rotacionado para esquerda.

# Balanceamento de Árvores AVL

## ◆ Tipos de rotação

### ◆ 2º Caso: Rotação simples à direita

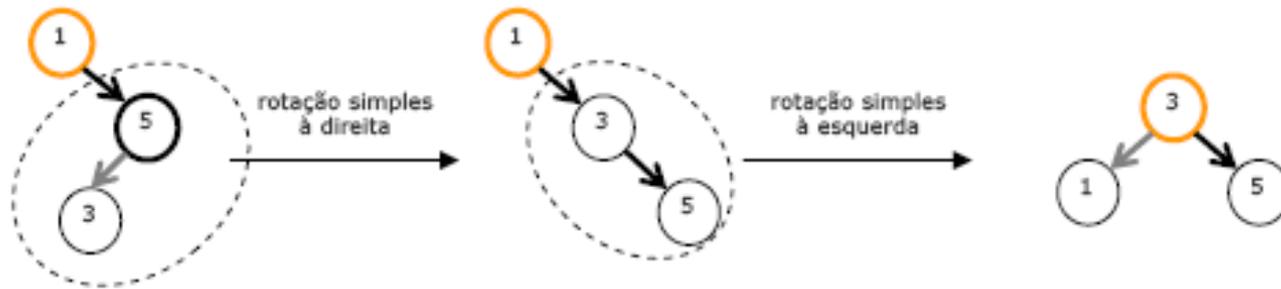


- O nó intermediário(3) é maior que seu pai(1), porém, é menor que seu filho(5).
- Logo o nó intermediário(3) deve ser escolhido para ser a raiz da árvore resultante.
- O nó corrente(1) tem que cair para a direita, ou seja, ser rotacionado para direita.

# Balanceamento de Árvores AVL

## Tipos de rotação

### 3º Caso: Rotação dupla à esquerda



O nó intermediário(5) é maior que seu pai(1) e é maior que seu filho(3).

Logo o nó intermediário(5) deve ser trocado com o seu filho(3), por uma rotação à direita, caindo no 2º caso; Então aplica-se uma rotação à esquerda.



# Balanceamento de Árvores AVL

- ◆ Uma **árvore AVL** pode se tornar desbalanceada em 4 situações, mas somente duas são analisadas pois há simetria entre os casos.
  - ◆ Inserção de um nó na árvore esquerda do filho esquerdo
  - ◆ Inserção de um nó na árvore direita do filho esquerdo
  - ◆ Inserção de um nó na árvore esquerda do filho direito
  - ◆ Inserção de um nó na árvore direita do filho direito
- ◆ Casos 1 e 4 simétricos; casos 2 e 3 simétricos
- ◆ Casos 1 e 4 são resolvidos com uma rotação simples.
- ◆ Casos 2 e 3 são resolvidos com uma rotação dupla.

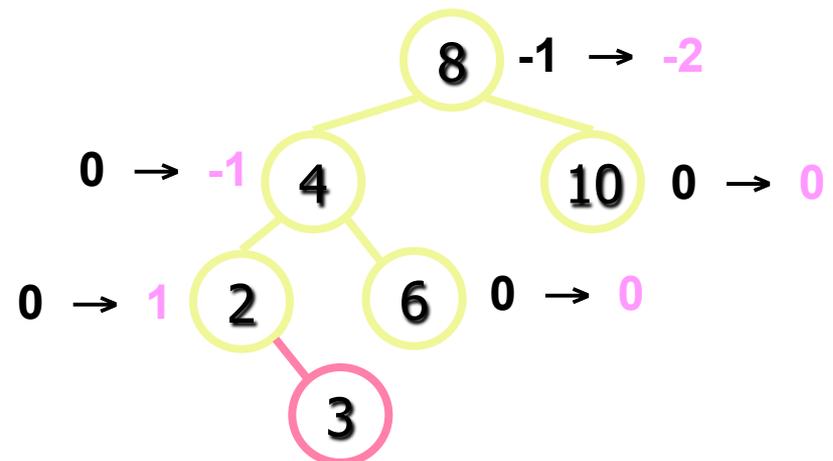
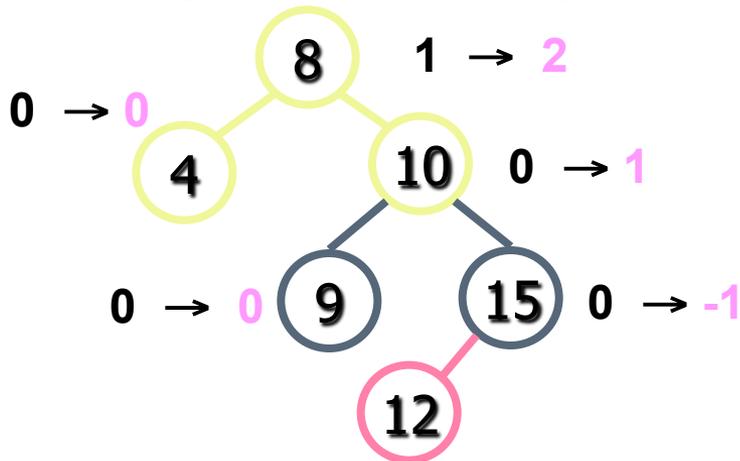
# Balanceamento de Árvores AVL

- ◆ **Balanceamento tipo 1:**

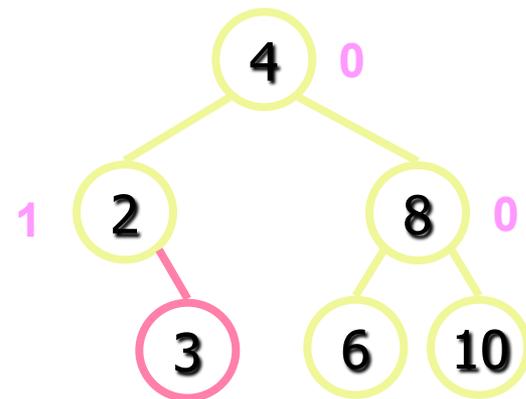
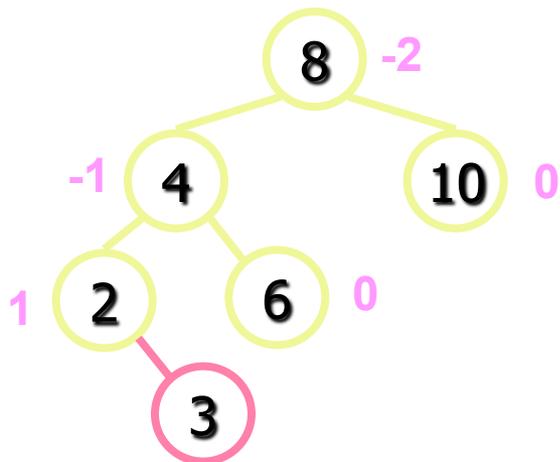
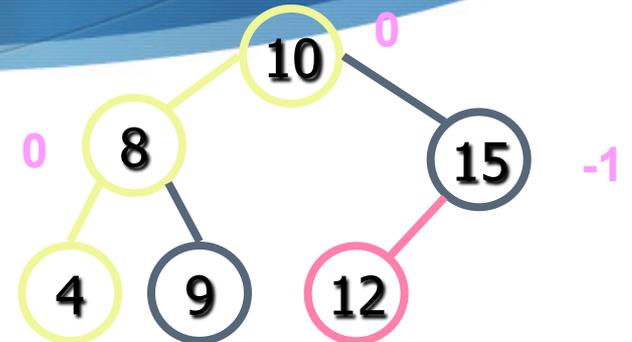
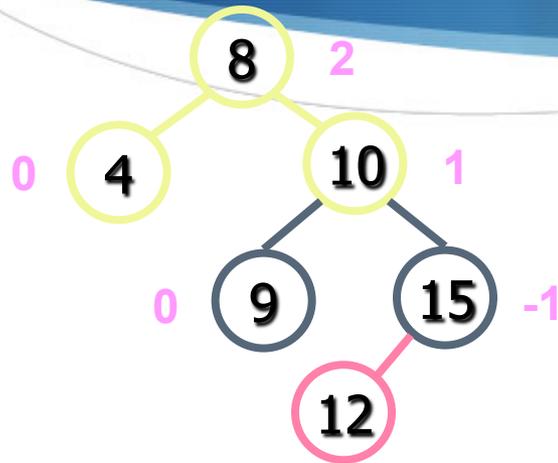
- ◆ o nó raiz de uma subárvore tem fator de balanceamento 2 (-2) e tem um filho com fator de balanceamento 1 (-1) o qual tem o mesmo sinal que o fator de balanceamento do nó pai.

- ◆ **Solução:** rotação simples sobre o nó de fator de balanceamento 2 (-2).

- ◆ Rotações são feitas à esquerda quando o fator de balanceamento for positivo e à direita quando o fator de balanceamento for negativo.



# Balanceamento de Árvores AVL



# Balanceamento de Árvores AVL

- Algoritmo para rotação à direita sobre o nó p

```
rotacao_direita ( p );  
{  
  q = esq(p);  
  temp = dir(q);  
  dir(q) = p;  
  esq(p) = temp;  
}
```

- Algoritmo para rotação à esquerda sobre o nó p

```
rotacao_esquerda(p)  
{  
  q = dir(p);  
  temp = esq(q);  
  esq(q) = p;  
  dir(p) = temp;  
}
```

# Balanceamento de Árvores AVL

## ◆ Balanceamento tipo 2:

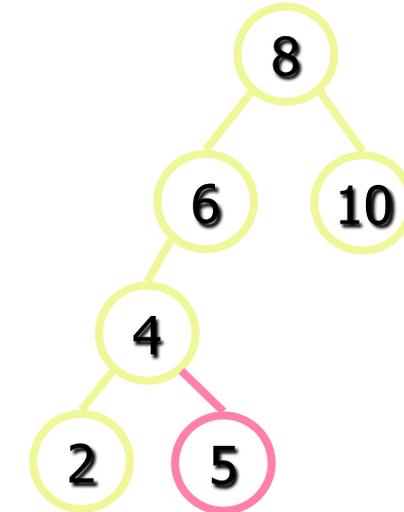
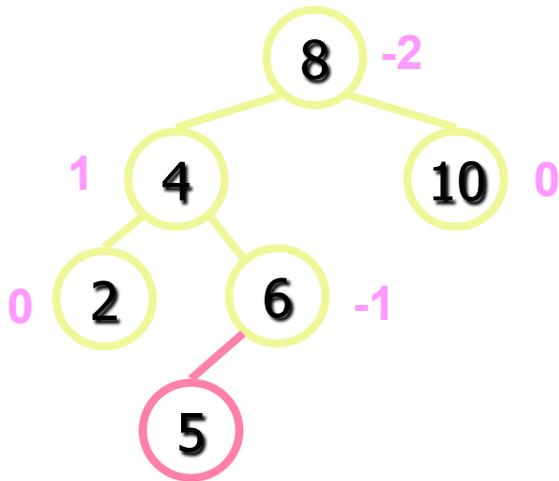
- ◆ o nó raiz de uma subárvore tem fator de balanceamento 2 (ou -2) e tem um filho com fator de balanceamento -1 (ou 1) o qual tem o sinal oposto ao fator de balanceamento do nó pai.

## ◆ Solução: rotação dupla.

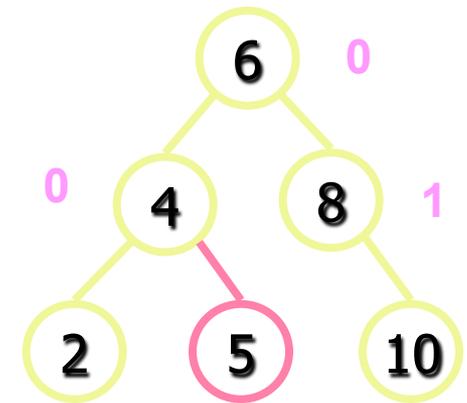
- ◆ primeiro roda-se o nó com fator de balanceamento 1 (-1) na direção apropriada.

- ◆ depois roda-se o nó que tinha fator de balanceamento -2 (2) na direção oposta.

# Balanceamento de Árvores AVL



Rotação à esquerda



Rotação à direita

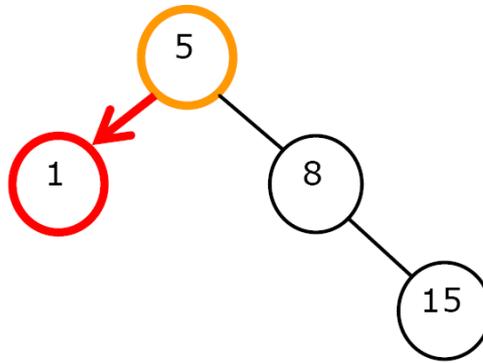
# Árvores AVL

## ◆ Remoção

- ◆ A remoção em árvores AVL é similar à remoção em uma árvore binária de busca.
- ◆ Todavia, é preciso verificar o balanceamento e, se necessário, aplicar algumas das rotações.
- ◆ A retirada de um valor de uma Árvore AVL é mais complexa do que a inserção.
- ◆ Enquanto qualquer inserção sempre ocorre numa folha da árvore, uma retirada envolve pelo menos dois casos distintos: os nós com dois filhos e os demais nós.
- ◆ Em geral é usada a remoção por cópia pois a remoção por fusão pode criar um grande desbalanceamento na árvore.
- ◆ A remoção de um nó não precisa necessariamente de uma rotação imediata pois ela pode melhorar o FB de seu ascendente, mas ela pode piorar o FB do avô.

# Árvores AVL

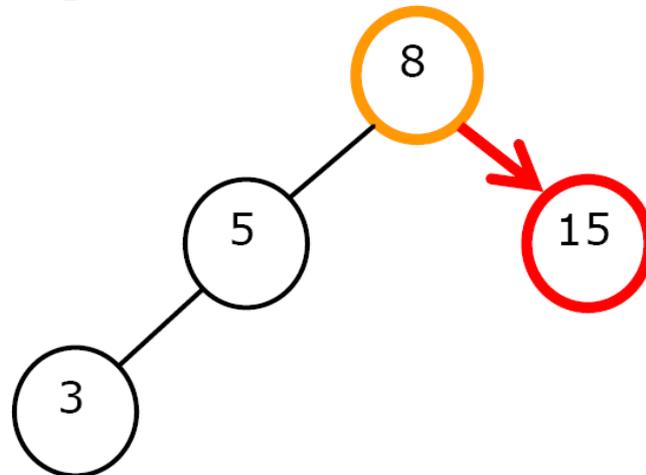
- ◆ Para a remoção pode-se cair em um dos casos:
- ◆ Rotação simples à esquerda:
  - ◆ A remoção do nó 1 deixa a árvore com desbalanceamento do tipo 1.
  - ◆ Aplica-se o algoritmo de rotação simples à esquerda sobre a raiz onde ocorreu a quebra do balanceamento.



# Árvores AVL

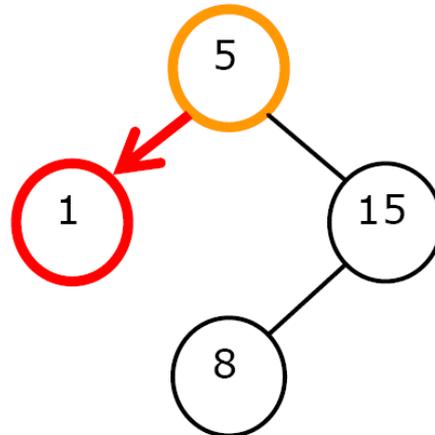
- ◆ Rotação simples à direita:

- ◆ A remoção do nó 15 deixa a árvore com desbalanceamento do tipo 2.
- ◆ Aplica-se o algoritmo de rotação simples à direita sobre a raiz onde ocorreu a quebra do balanceamento.



# Árvores AVL

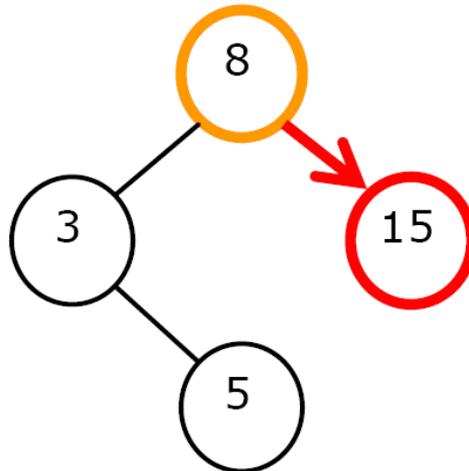
- ◆ Rotação dupla à esquerda:
  - ◆ A remoção do nó 1 deixa a árvore com desbalanceamento do tipo 3.
  - ◆ Aplica-se o algoritmo de rotação dupla à esquerda sobre a raiz onde ocorreu a quebra do balanceamento.



# Árvores AVL

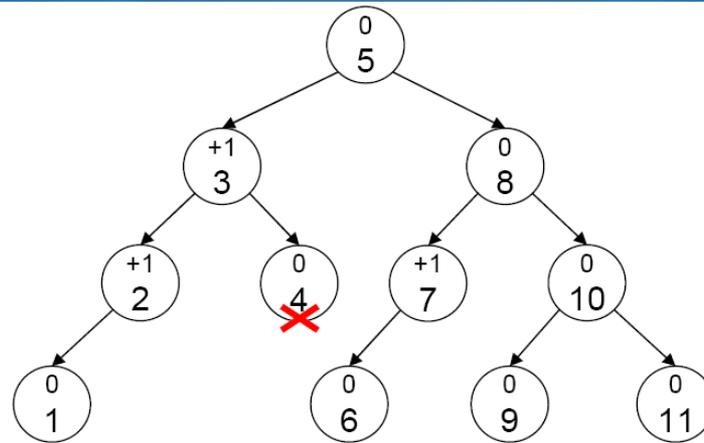
- ◆ Rotação dupla à direita:

- ◆ A remoção do nó 15 deixa a árvore com desbalanceamento do tipo 4.
- ◆ Aplica-se o algoritmo de rotação dupla à direita sobre a raiz onde ocorreu a quebra do balanceamento.

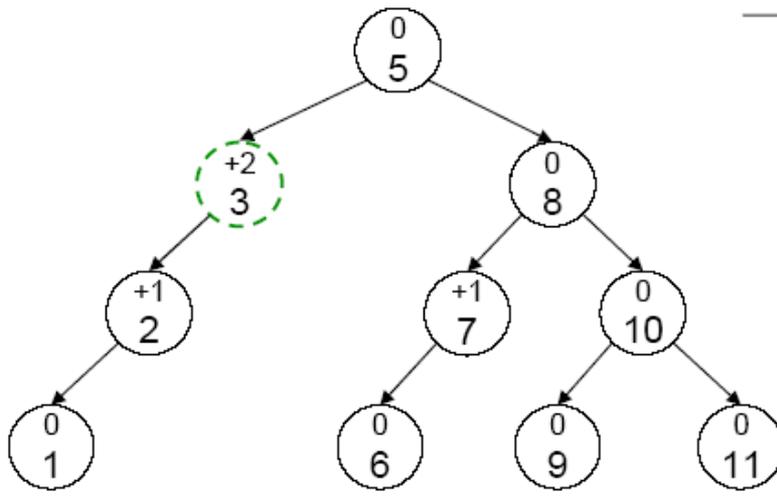


# Árvores AVL

Antes da remoção

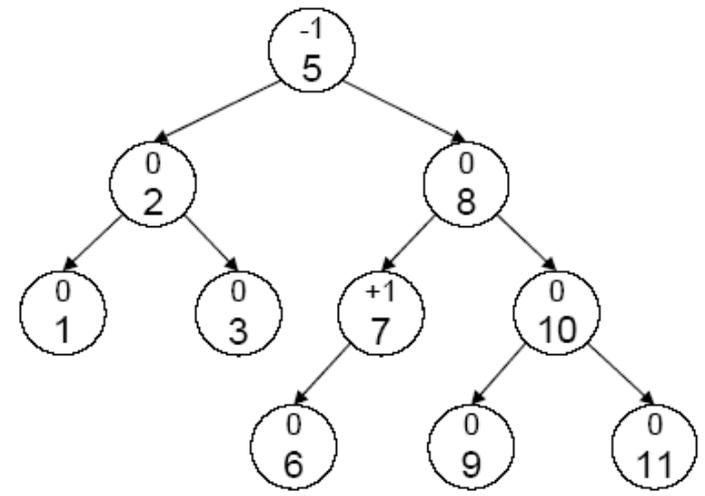


Após a remoção



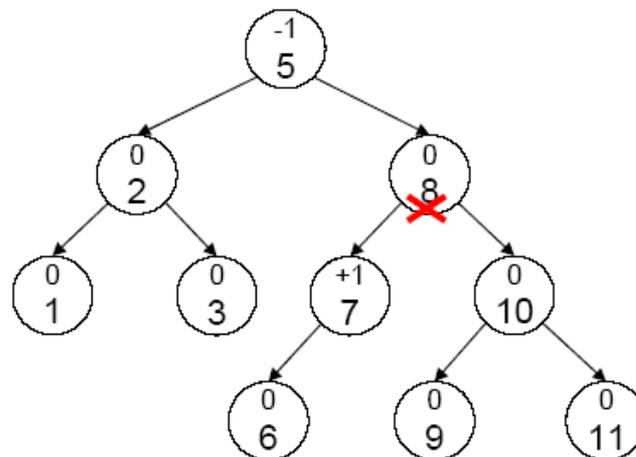
LL →

Após o rebalanceamento

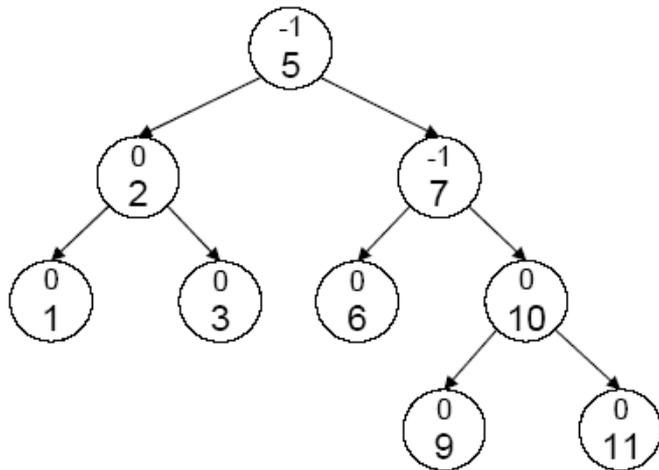


# Árvores AVL

Antes da remoção



Após a remoção

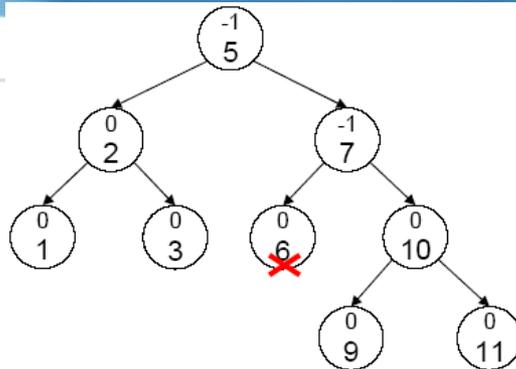


Apos o rebalanceamento

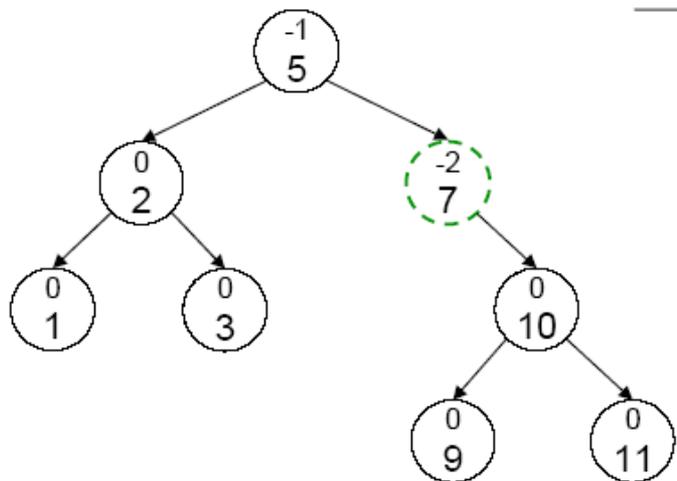
Sem necessidade de rebalanceamento

# Árvores AVL

Antes da remoção

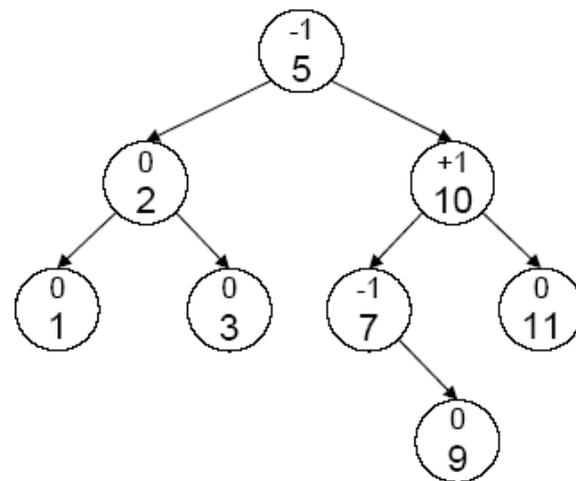


Após a remoção



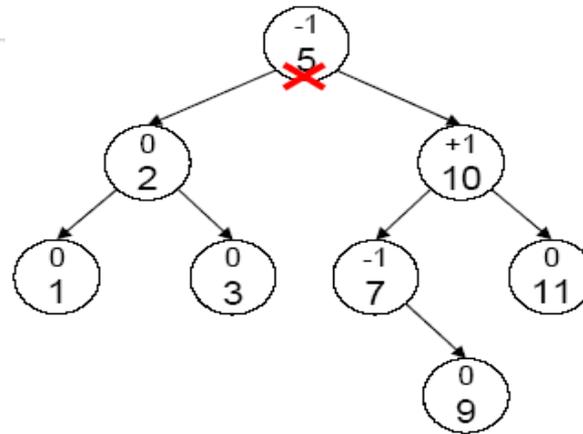
RR →

Após o rebalanceamento

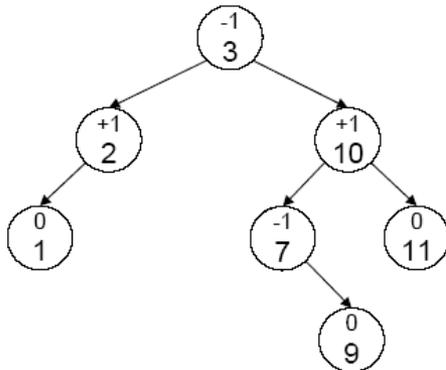


# Árvores AVL

Antes da remoção



Após a remoção

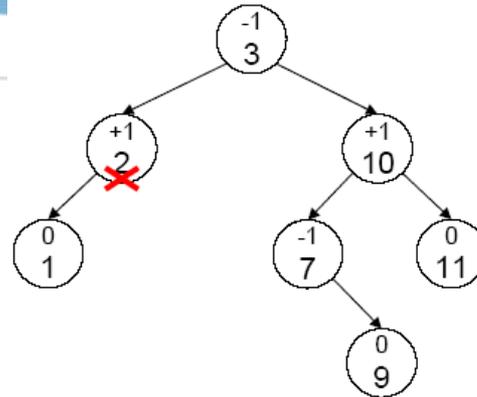


Após o rebalanceamento

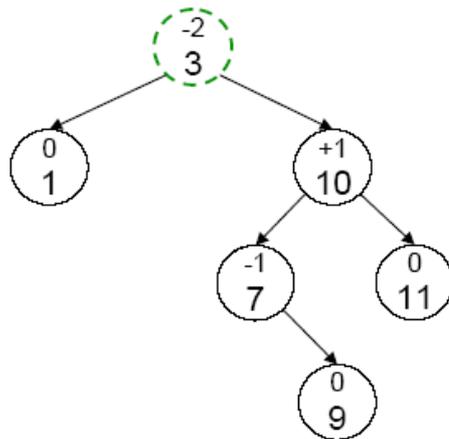
Sem necessidade de rebalanceamento

# Árvores AVL

Antes da remoção

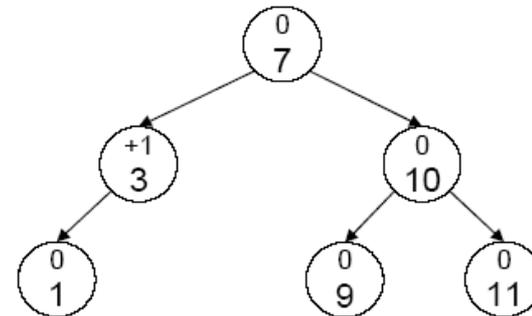


Após a remoção



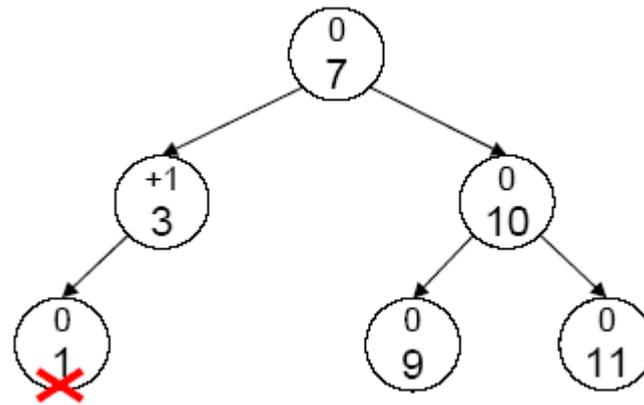
RL →

Após o rebalanceamento

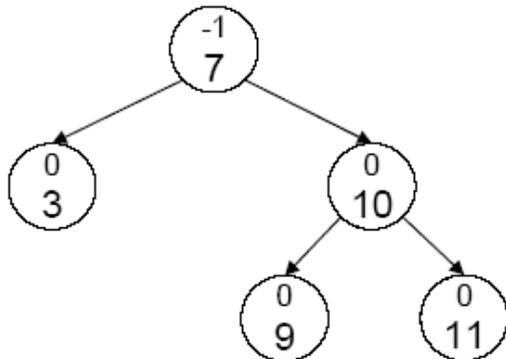


# Árvores AVL

Antes da remoção



Após a remoção

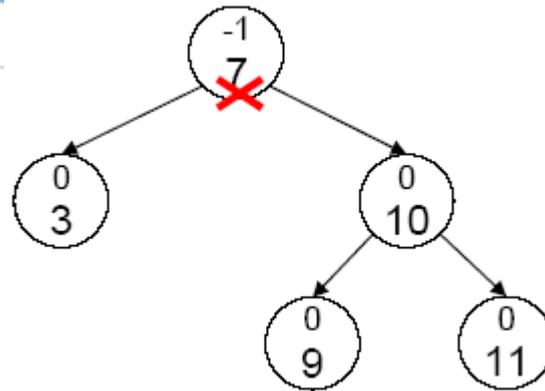


Após o rebalanceamento

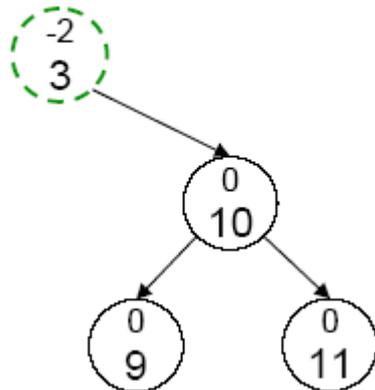
Sem necessidade de rebalanceamento

# Árvores AVL

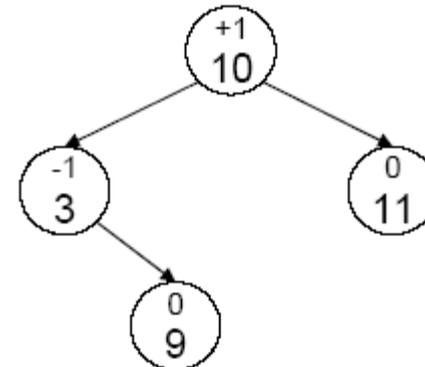
Antes da remoção



Após a remoção

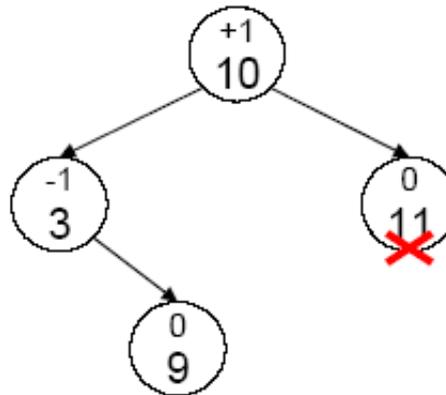


Após o rebalanceamento

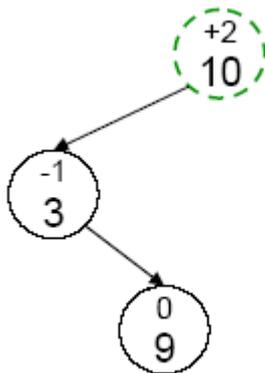


# Árvores AVL

Antes da remoção



Após a remoção



Após o rebalanceamento

