



Algoritmos e Estrutura de Dados Avançado

Encontro 07 – Árvores AVL



Materiais e dúvidas



Nenhum material será enviado via e-mail.
Os materiais serão disponibilizados no **AVA** e no disco virtual: bit.ly/edauniderp



Dúvidas, questionamentos, entre outros
deverão ser realizados **APENAS** pelo
e-mail e **AVA**.



Para ingressar no grupo do **Whatsapp** da
disciplina acesse o link linklist.bio/noiza e
selecione sua disciplina.



Árvores

Árvores AVL

Árvore AVL

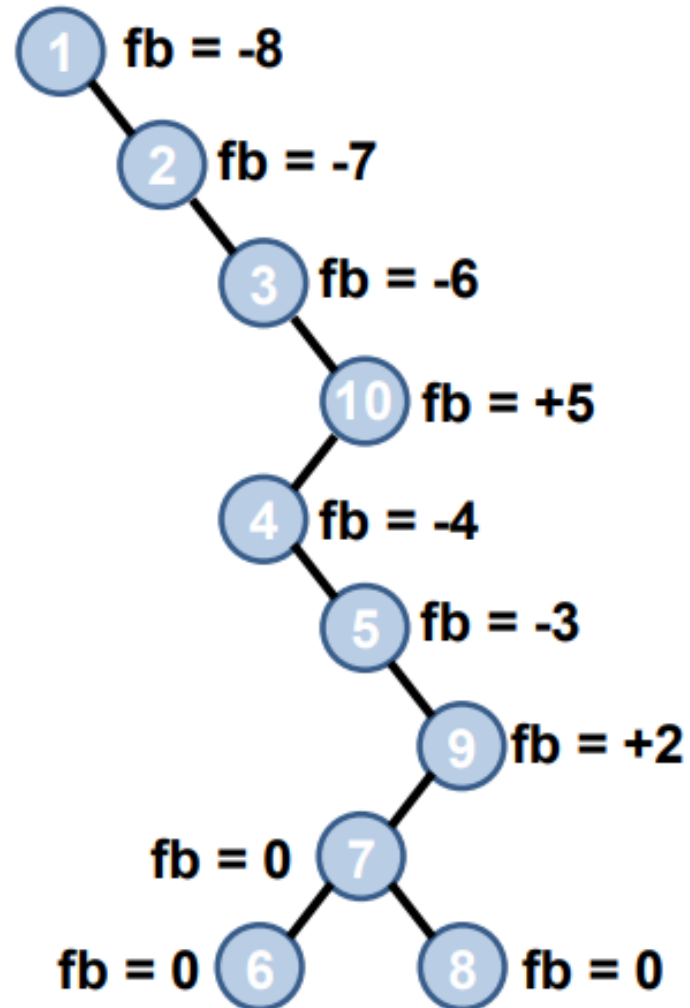
- A eficiência da busca em uma árvore binária depende do seu balanceamento.
 - $O(\log N)$, se a árvore está balanceada
 - $O(N)$, se a árvore não está balanceada
- N corresponde ao número de nós na árvore

Problema do balanceamento

- Infelizmente, os algoritmos de inserção e remoção em árvores binárias não garantem que a árvore gerada a cada passo esteja balanceada.
- Dependendo da ordem em que os dados são inseridos na árvore, podemos criar uma árvore na forma de uma escada

Problema do balanceamento

Inserção dos valores
{1,2,3,10,4,5,9,7,8,6}



Solução para o Problema do balanceamento

- Modificar as operações de inserção e remoção de modo a balancear a árvore a cada nova inserção ou remoção.
- Garantir que a diferença de alturas das subárvores esquerda e direita de cada nó seja de no máximo uma unidade
- Exemplos de árvores balanceadas
 - Árvore AVL
 - Árvore 2-3-4
 - Árvore Rubro-Negra

Árvore AVL

Definição:

- Tipo de árvore binária balanceada com relação à altura das suas subárvores
- Criada por Adelson-Velskii e Landis, de onde recebeu a sua nomenclatura, em 1962

Árvore AVL

Definição:

- Permite o rebalanceamento local da árvore
 - Apenas parte afetada pela inserção ou remoção é rebalanceada
- Usa rotações simples ou duplas na etapa de rebalanceamento
 - Executadas a cada inserção ou remoção
 - As rotações buscam manter a árvore binária como uma árvore quase completa
 - Custo máximo de qualquer algoritmo é $O(\log N)$

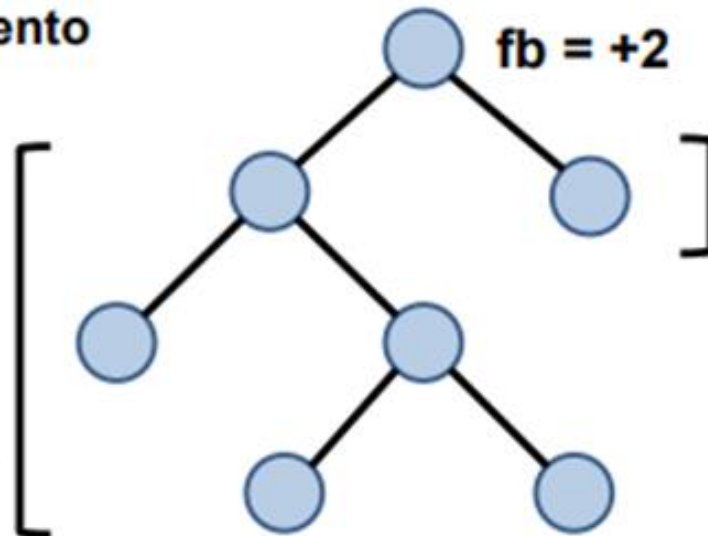
Árvore AVL

Objetivo das rotações:

- Corrigir o fator de balanceamento (ou fb), que é a diferença entre as alturas das subárvores de um nó
- Caso uma das subárvores de um nó não existir, então a altura dessa subárvore será igual a -1.

Fator de Balanceamento
 $FB = AE - AD$

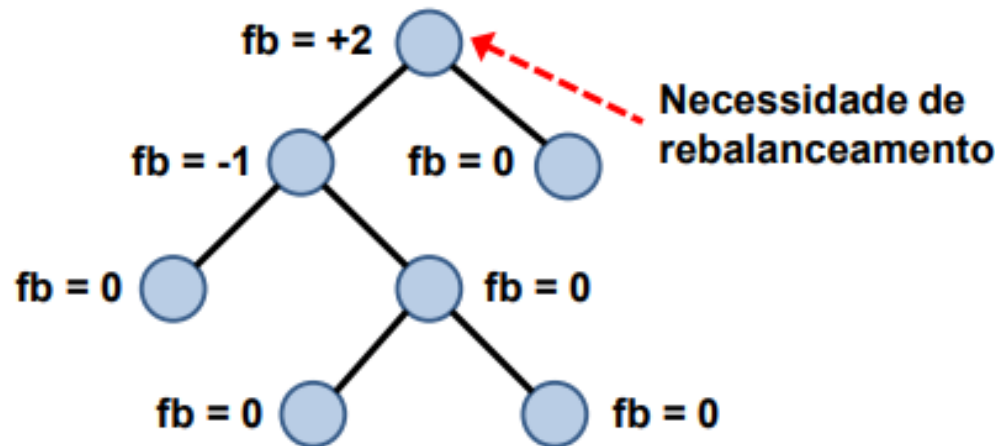
AE
Altura da
subárvore
ESQUERDA



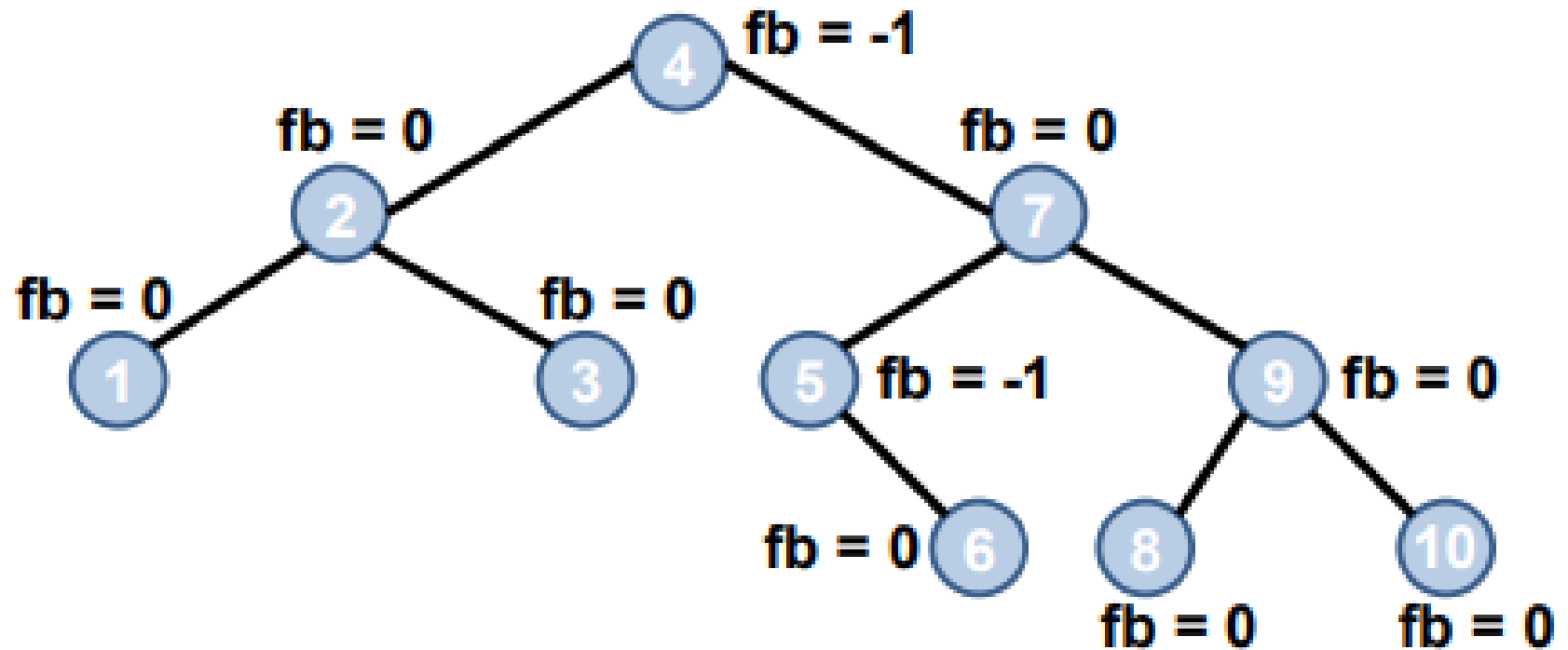
AD
Altura da
subárvore
DIREITA

Árvore AVL

- As alturas das subárvores de cada nó diferem de no máximo uma unidade
- O fator de balanceamento deve ser +1, 0 ou -1
- **Se $fb > +1$ ou $fb < -1$:** a árvore deve ser balanceada naquele nó

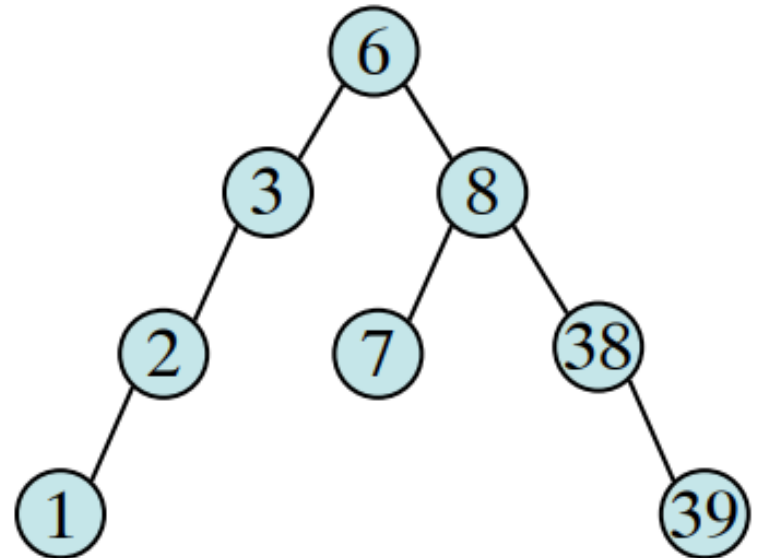
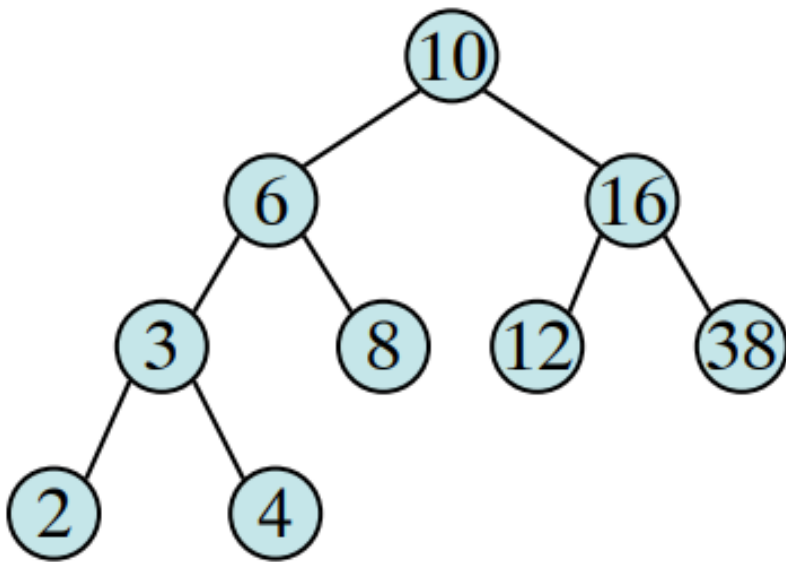


Árvore AVL



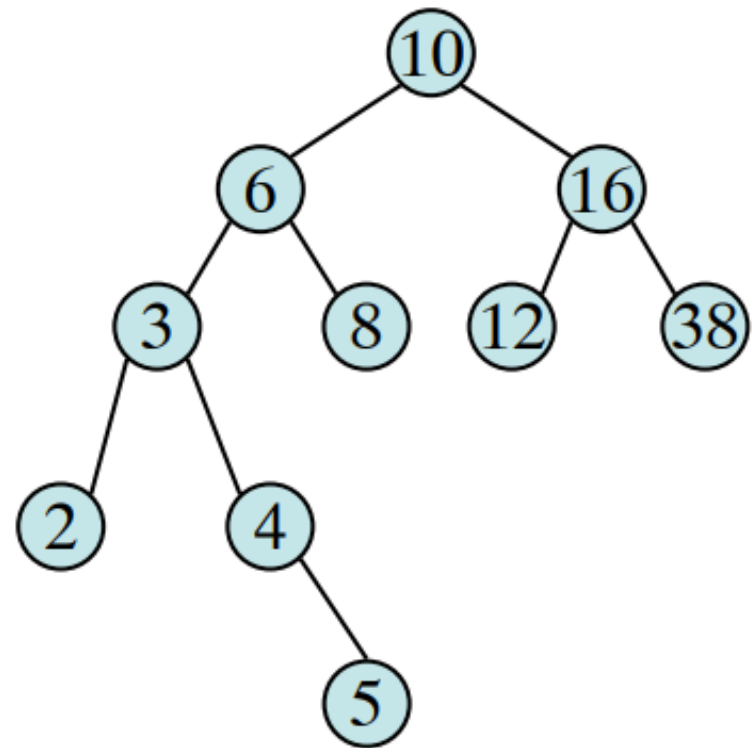
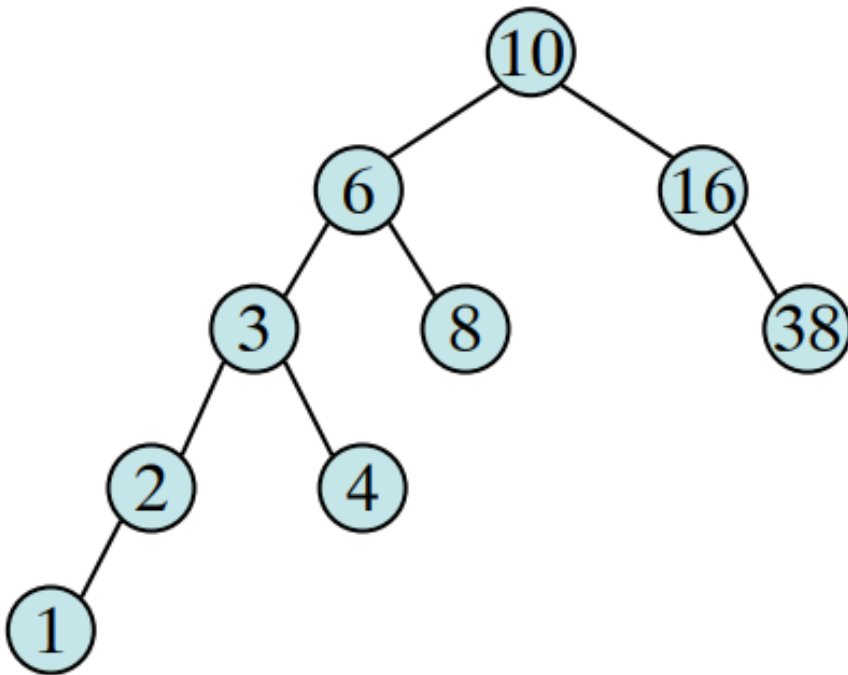
Árvore AVL

- Determinar o balanceamento de cada nó e dizer se a árvore é AVL



Árvore AVL

- Determinar o balanceamento de cada nó e dizer se a árvore é AVL



Árvores AVL

Rotações



Rotações

- Objetivo: corrigir o fator de balanceamento (fb) de cada nó
- Operação básica para balancear uma árvore AVL
- Ao todo, existem dois tipos de rotação
 - Rotação simples
 - Rotação dupla

Rotações

- As rotações diferem entre si pelo sentido da inclinação entre o nó pai e filho
- Rotação simples
 - O nó desbalanceado (pai), seu filho e o seu neto estão todos no mesmo sentido de inclinação
- Rotação dupla
 - O nó desbalanceado (pai) e seu filho estão inclinados no sentido inverso ao neto
 - Equivale a duas rotações simples

Rotações

- Ao todo, existem duas rotações simples e duas duplas:
 - Rotação simples a direita ou Rotação LL
 - Rotação simples a esquerda ou Rotação RR
 - Rotação dupla a direita ou Rotação LR
 - Rotação dupla a esquerda ou Rotação RL

Rotações

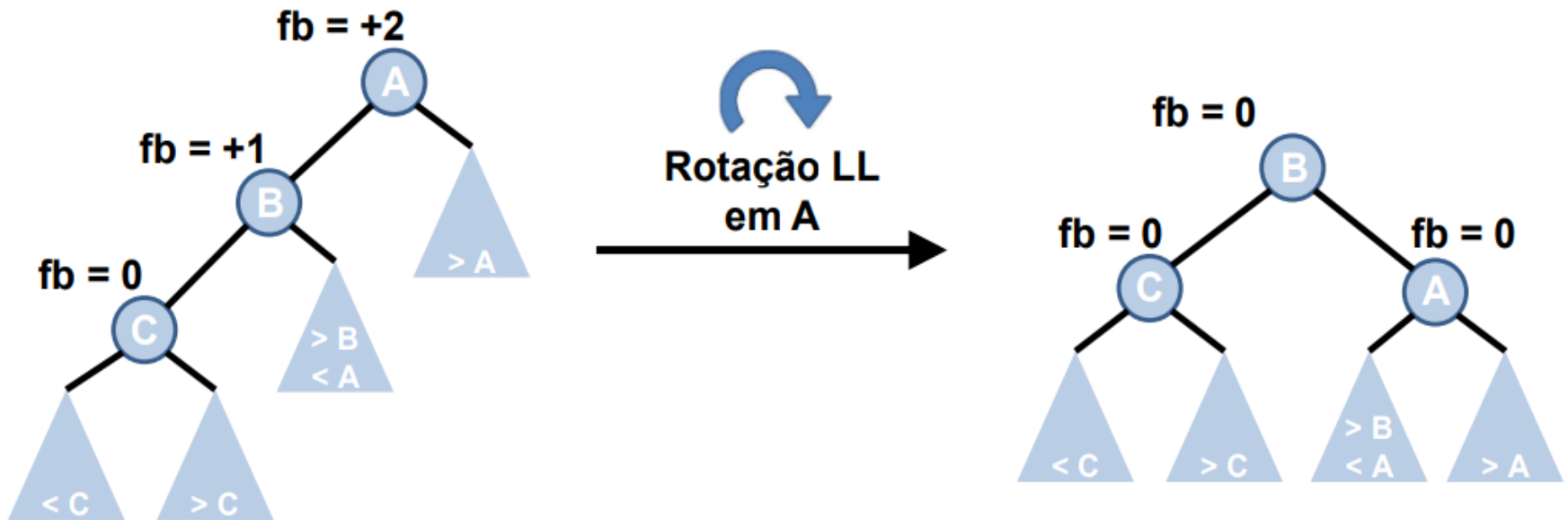
- Rotações são aplicadas no ancestral **mais próximo** do nó inserido cujo fator de balanceamento passa a ser +2 ou -2
- Após uma inserção ou remoção, devemos voltar pelo mesmo caminho da árvore e recalcular o fator de balanceamento (fb) de cada nó
- Se o fb desse nó for +2 ou -2, uma rotação deverá ser aplicada

Rotação LL

- Rotação LL ou rotação simples à direita
- Um novo nó é inserido na subárvore da esquerda do filho esquerdo de A
 - A é o nó desbalanceado
 - Dois elementos para a esquerda: **LEFT LEFT**
- É necessário fazer uma rotação à direita, de modo que o nó intermediário B ocupe o lugar de A, e A se torne a subárvore direita de B

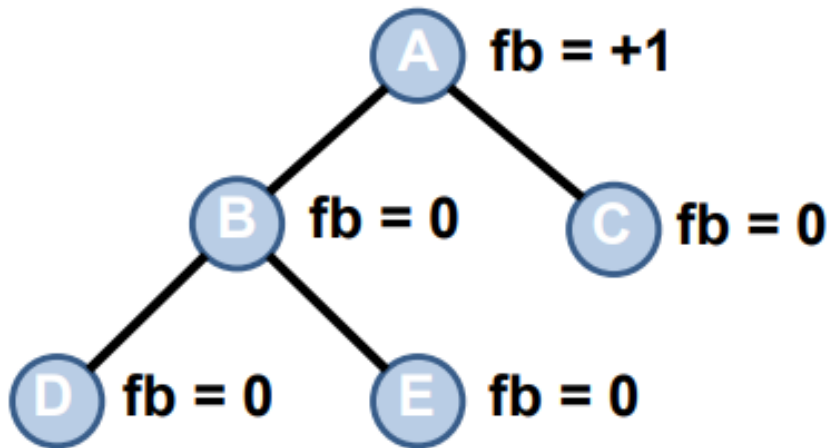
Rotação LL

- Exemplo



Rotação LL

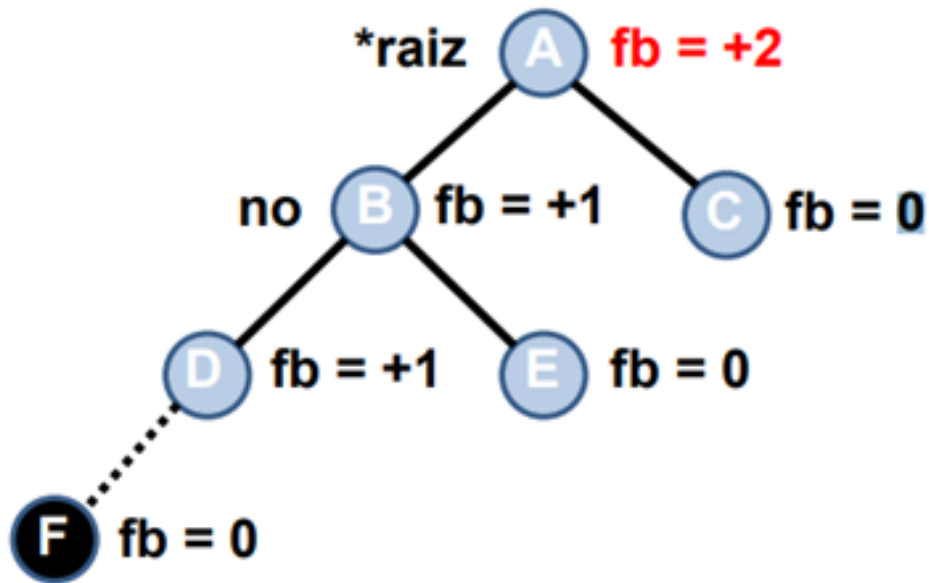
- Passo a passo



Árvore AVL e fator de balanceamento de cada nó

Rotação LL

- Passo a passo



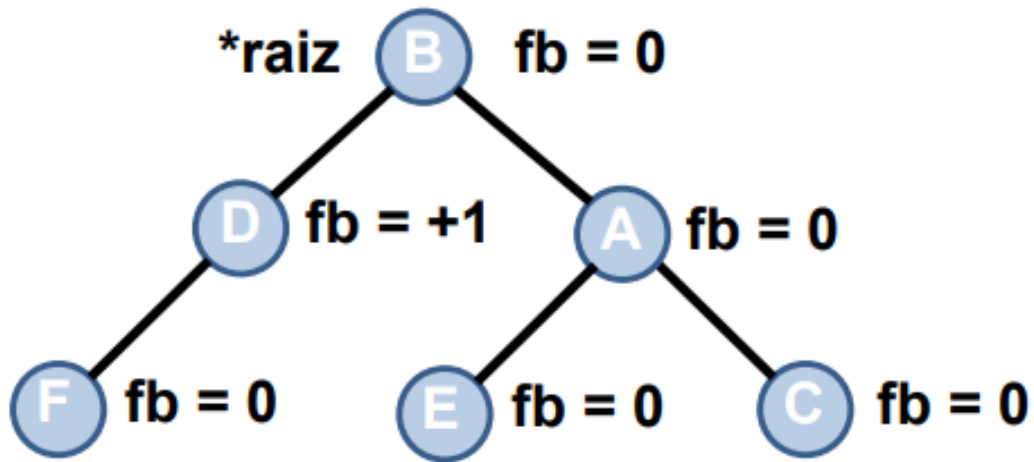
Inserção do nó F na árvore

Árvore fica desbalanceada no nó A.

Aplicar Rotação LL no nó A

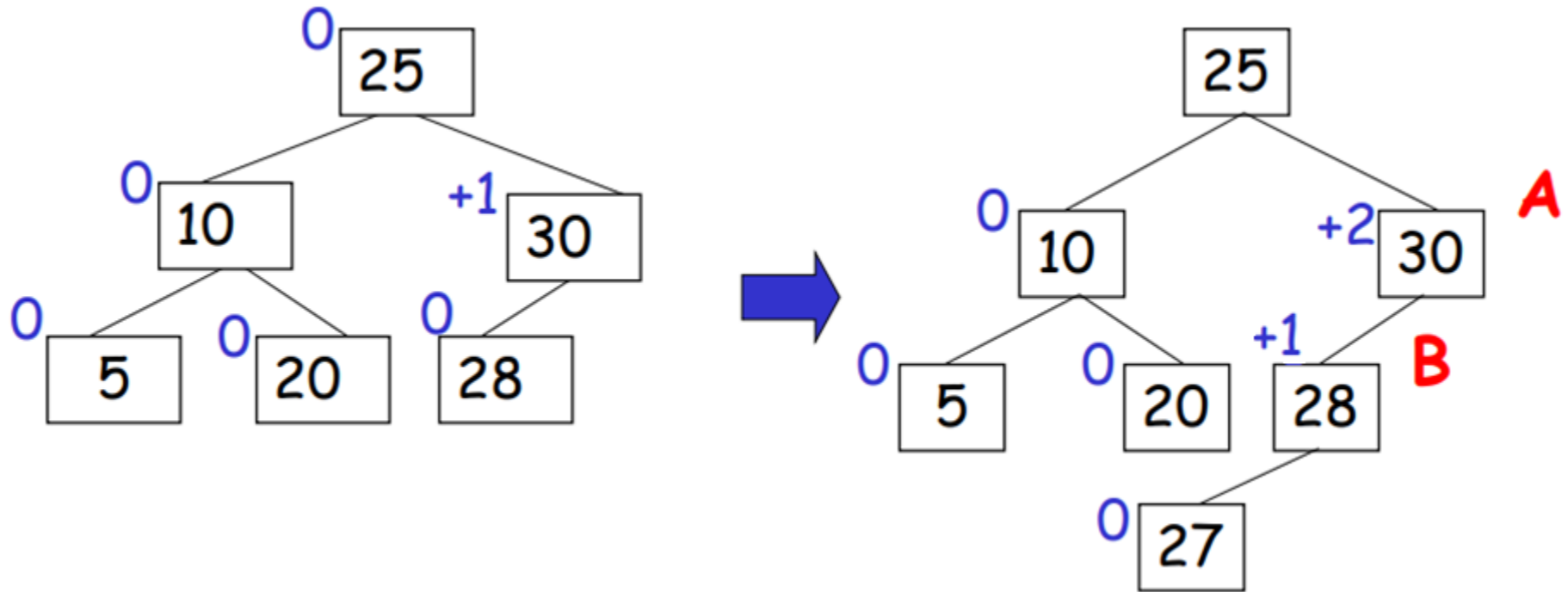
Rotação LL

- Passo a passo

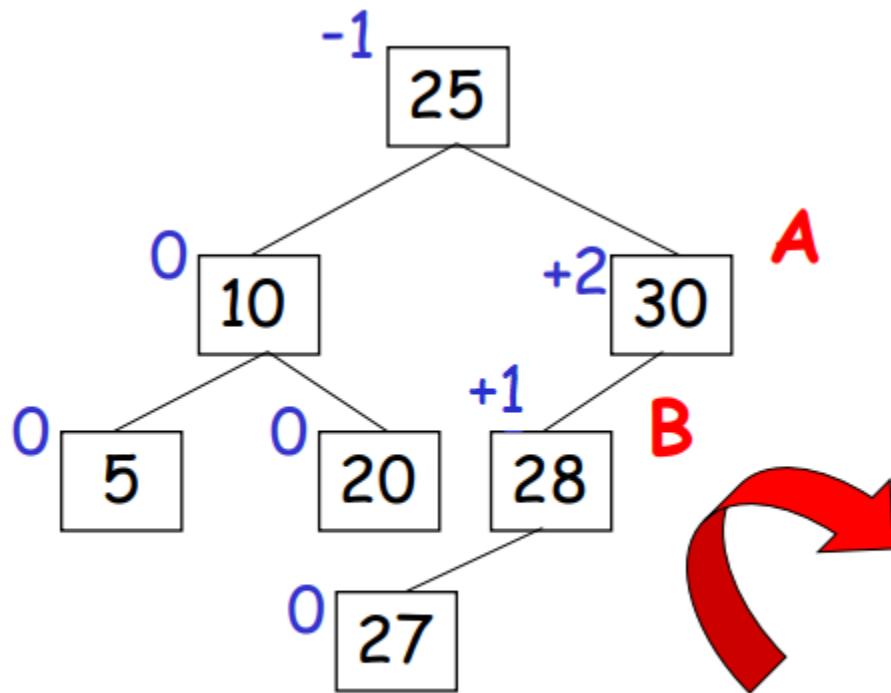


Árvore Balanceada

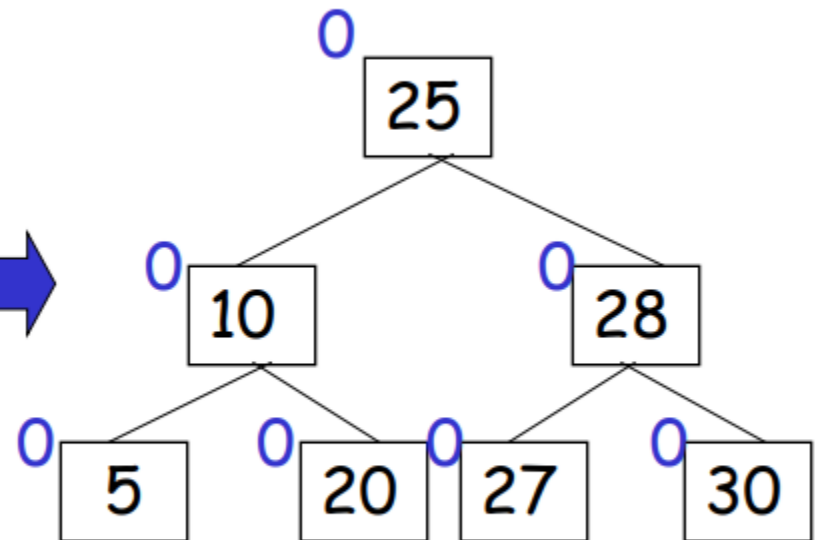
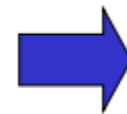
Rotação LL - Exemplo



Rotação LL - Exemplo



ANTES



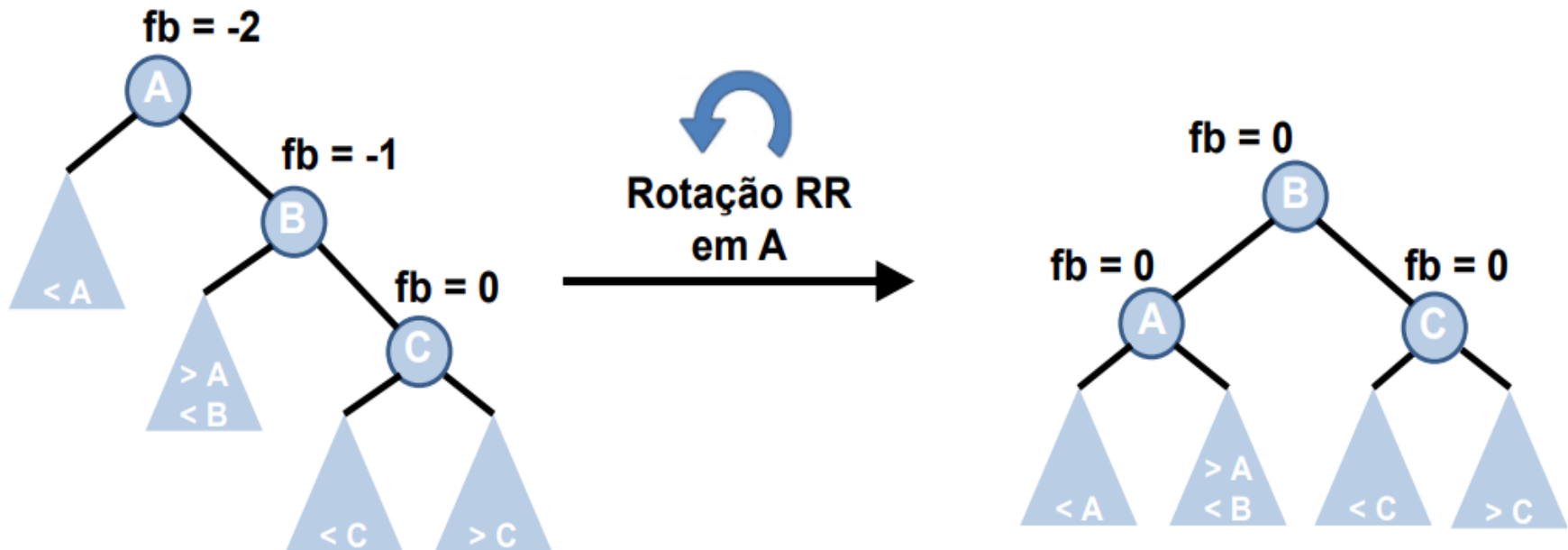
DEPOIS

Rotação RR

- Rotação RR ou rotação simples à esquerda
- Um novo nó é inserido na subárvore da direita do filho direito de A
 - A é o nó desbalanceado
 - Dois elementos para a direita: **RIGHT RIGHT**
- É necessário fazer uma rotação à esquerda, de modo que o nó intermediário B ocupe o lugar de A, e A se torne a subárvore esquerda de B

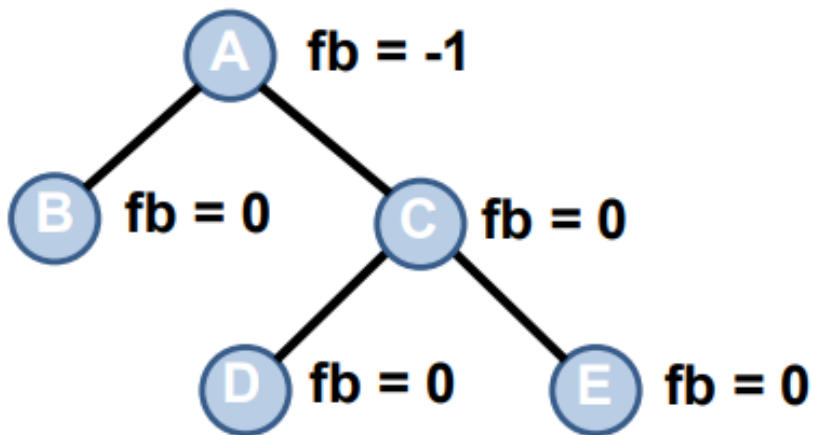
Rotação RR

- Exemplo



Rotação RR

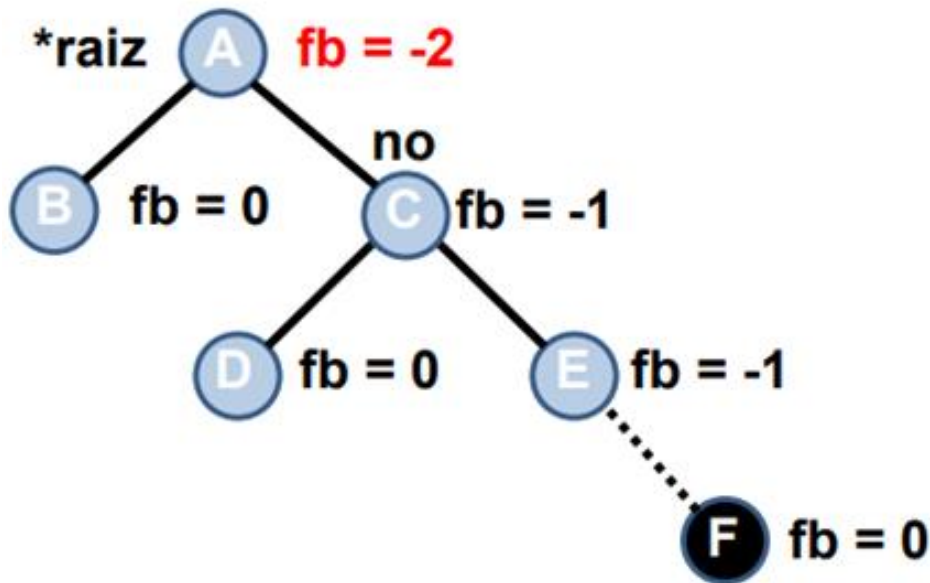
- Passo a passo



Árvore AVL e fator de balanceamento de cada nó

Rotação RR

- Passo a passo



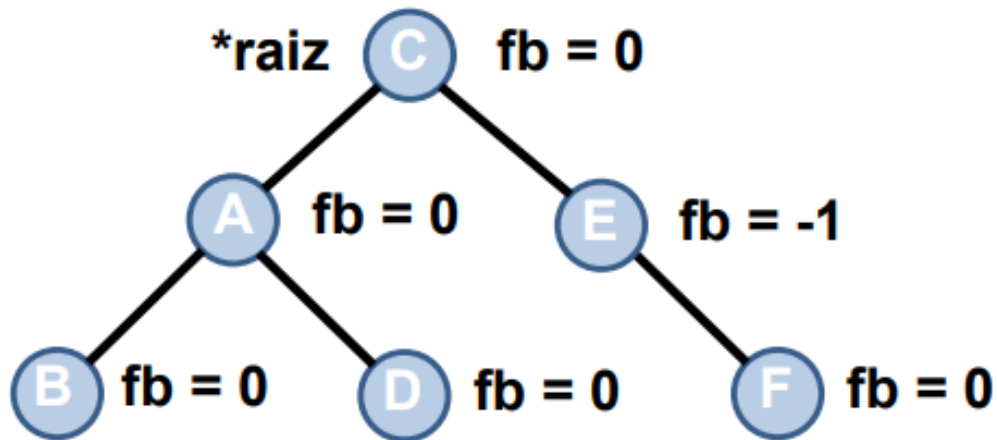
Inserção do nó F na árvore

Árvore fica desbalanceada no nó A.

Aplicar Rotação RR no nó A

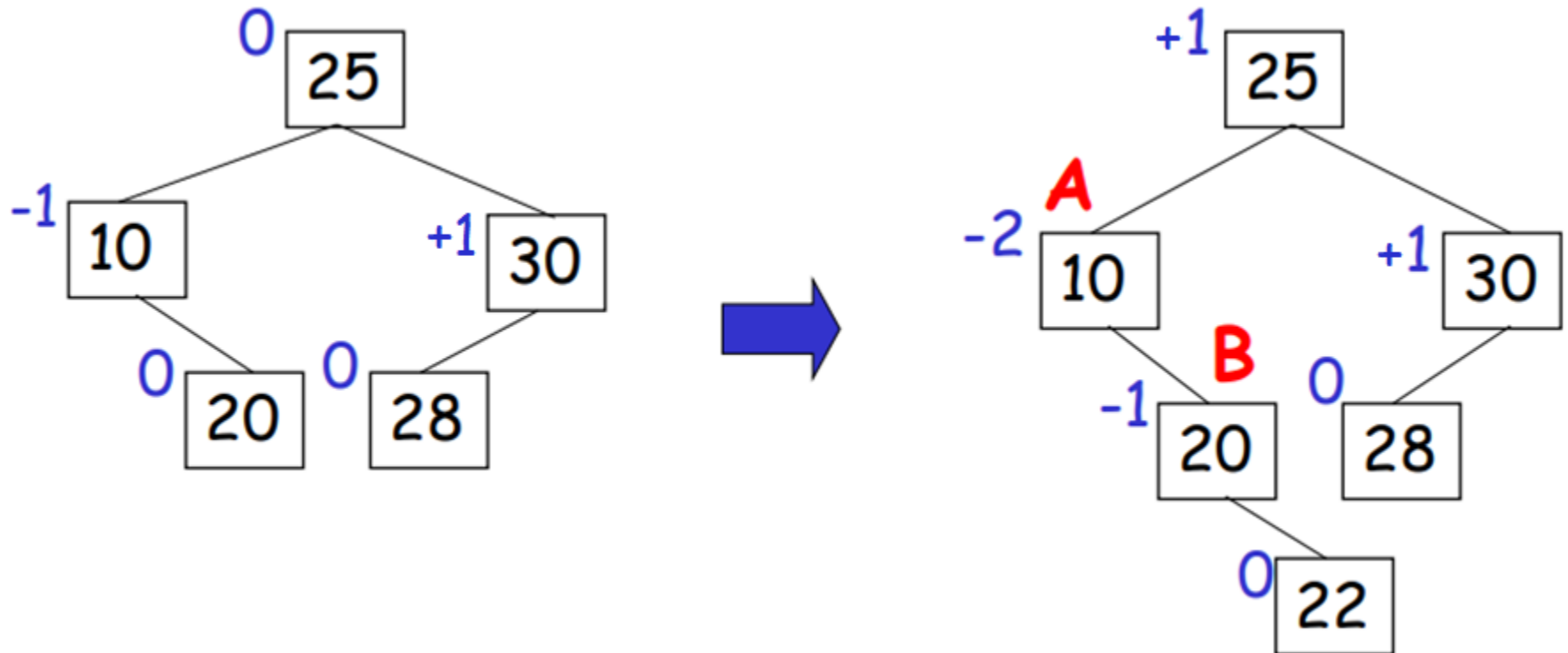
Rotação RR

- Passo a passo

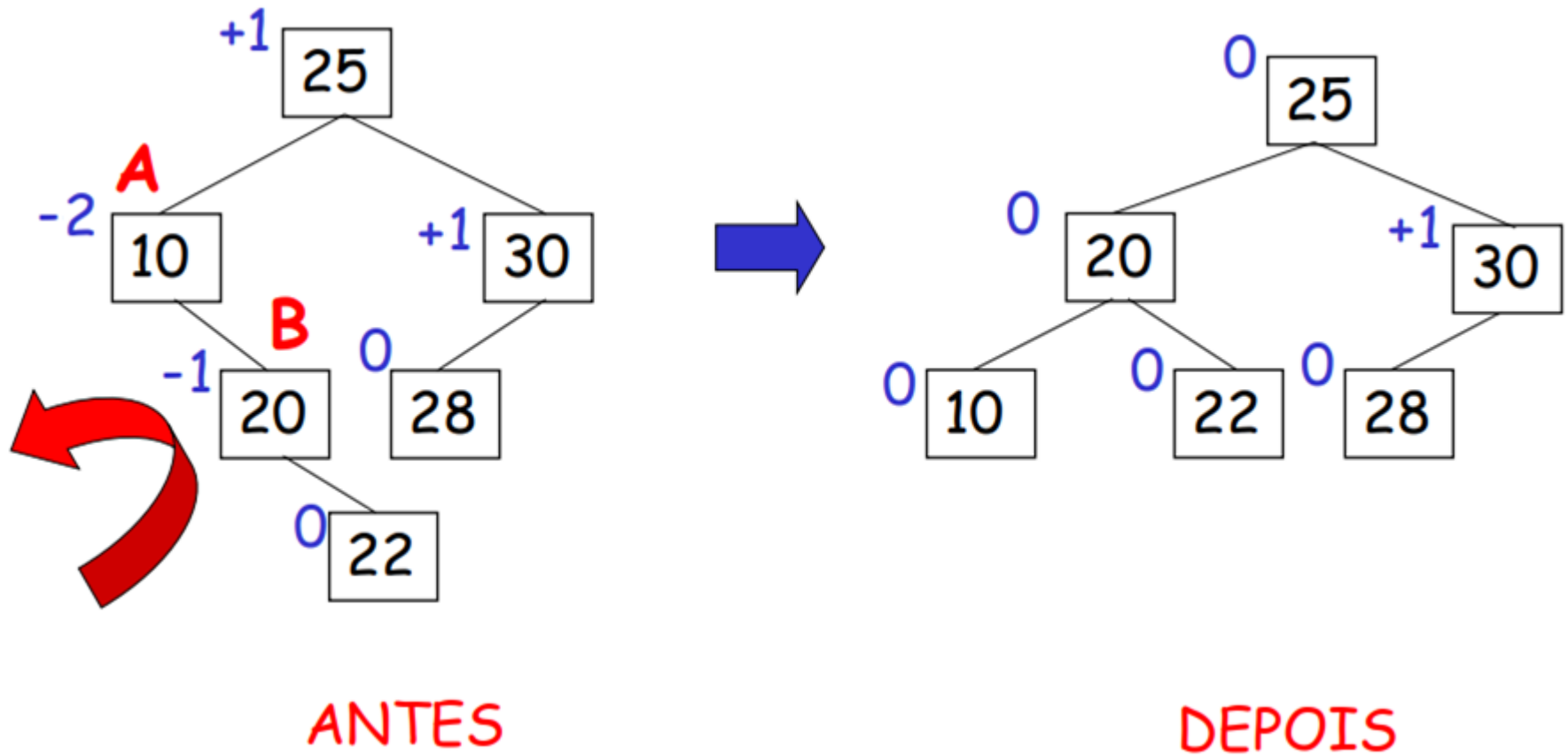


Árvore Balanceada

Rotação RR - Exemplo



Rotação RR - Exemplo

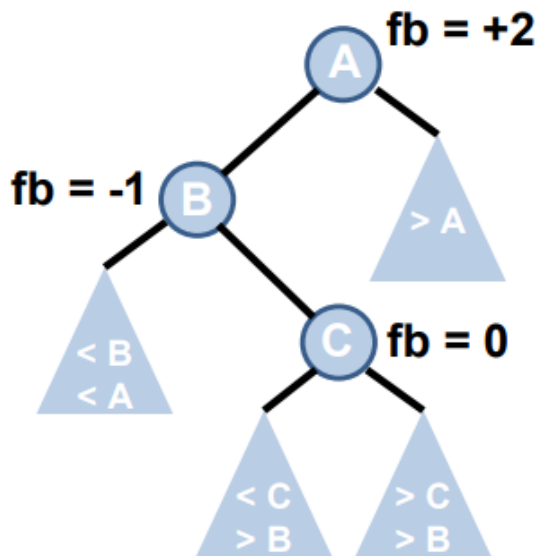


Rotação LR

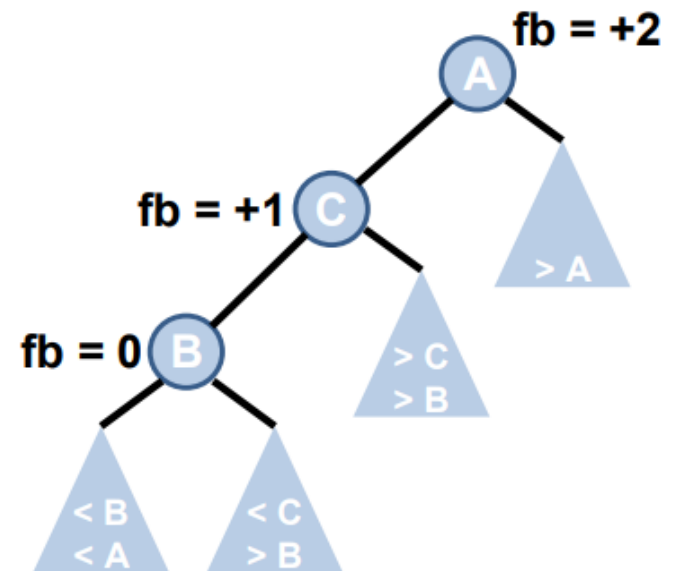
- Rotação LR ou rotação dupla à direita
- Um novo nó é inserido na subárvore da direita do filho esquerdo de A
 - A é o nó desbalanceado
 - Um elemento para a esquerda e outro para a direita: **LEFT RIGHT**
- É necessário fazer uma rotação dupla, de modo que o nó C se torne o pai dos nós A (filho da direita) e B (filho da esquerda)
 - Rotação RR em B
 - Rotação LL em A

Rotação LR

- Exemplo: primeira rotação

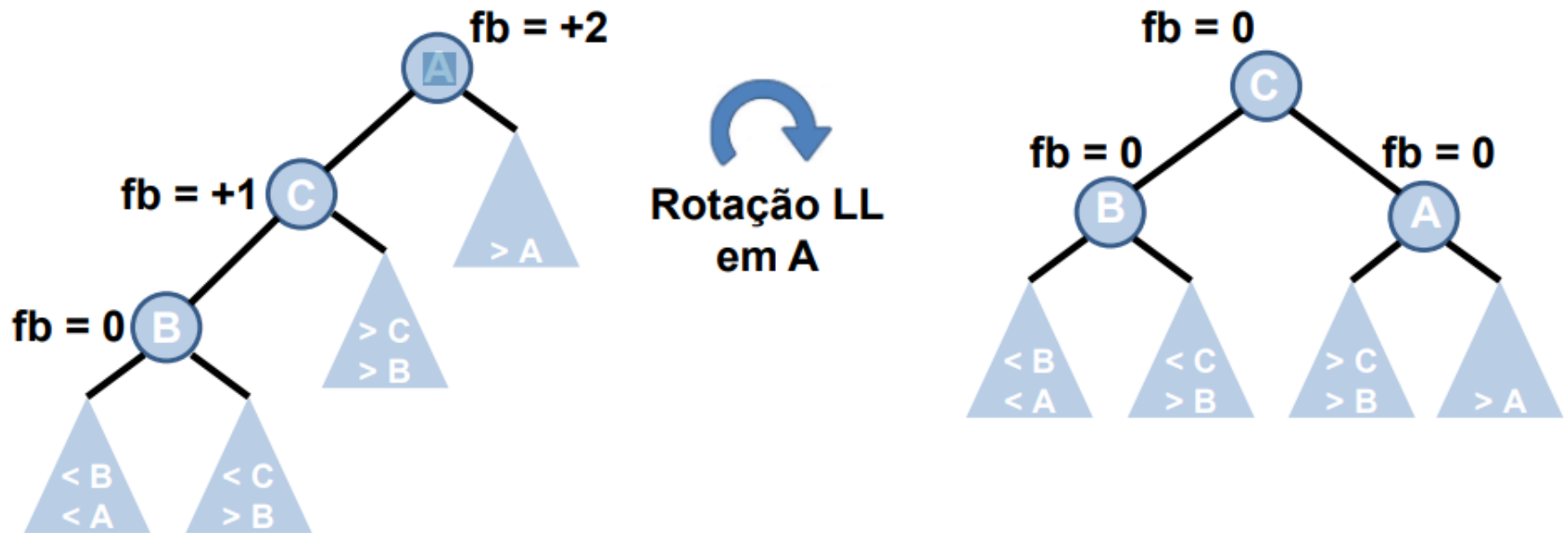



Rotação RR
em B



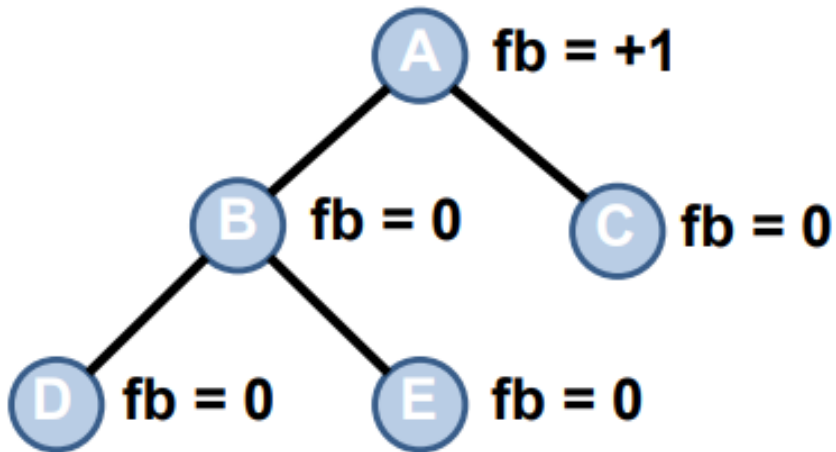
Rotação LR

- Exemplo: segunda rotação



Rotação LR

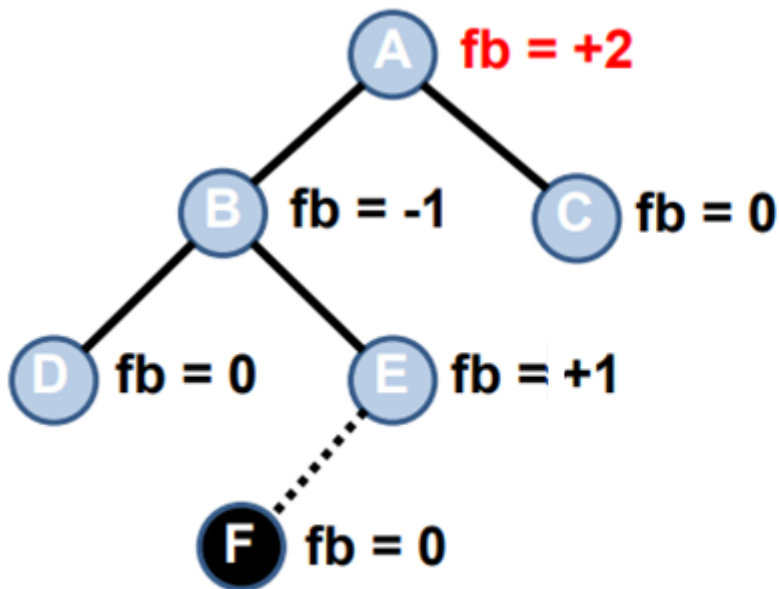
- Passo a passo



Árvore AVL e fator de balanceamento de cada nó

Rotação LR

- Passo a passo



Inserção do nó F na árvore

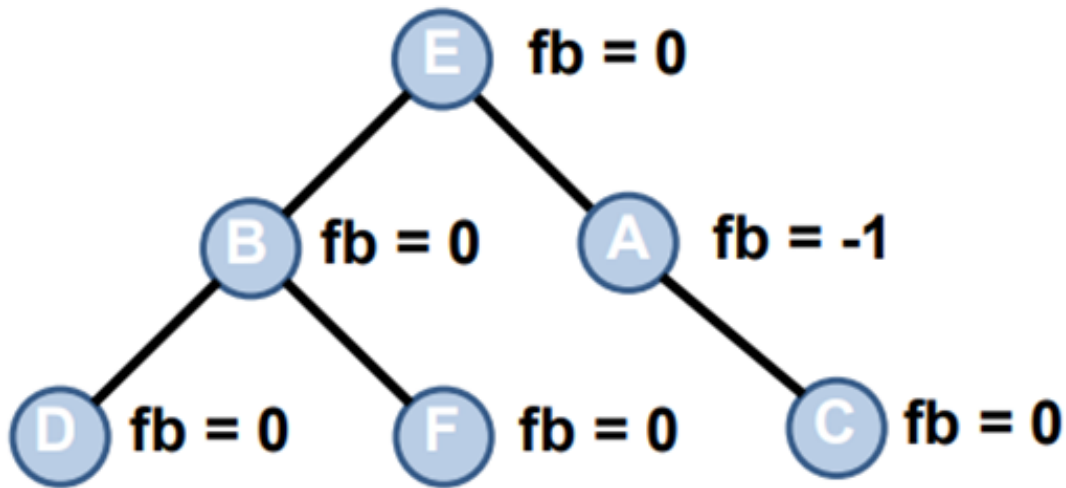
Árvore fica desbalanceada no nó A.

**Aplicar Rotação LR no nó A.
Isso equivale a:**

- **Aplicar a Rotação RR no nó B**
- **Aplicar a Rotação LL no nó A**

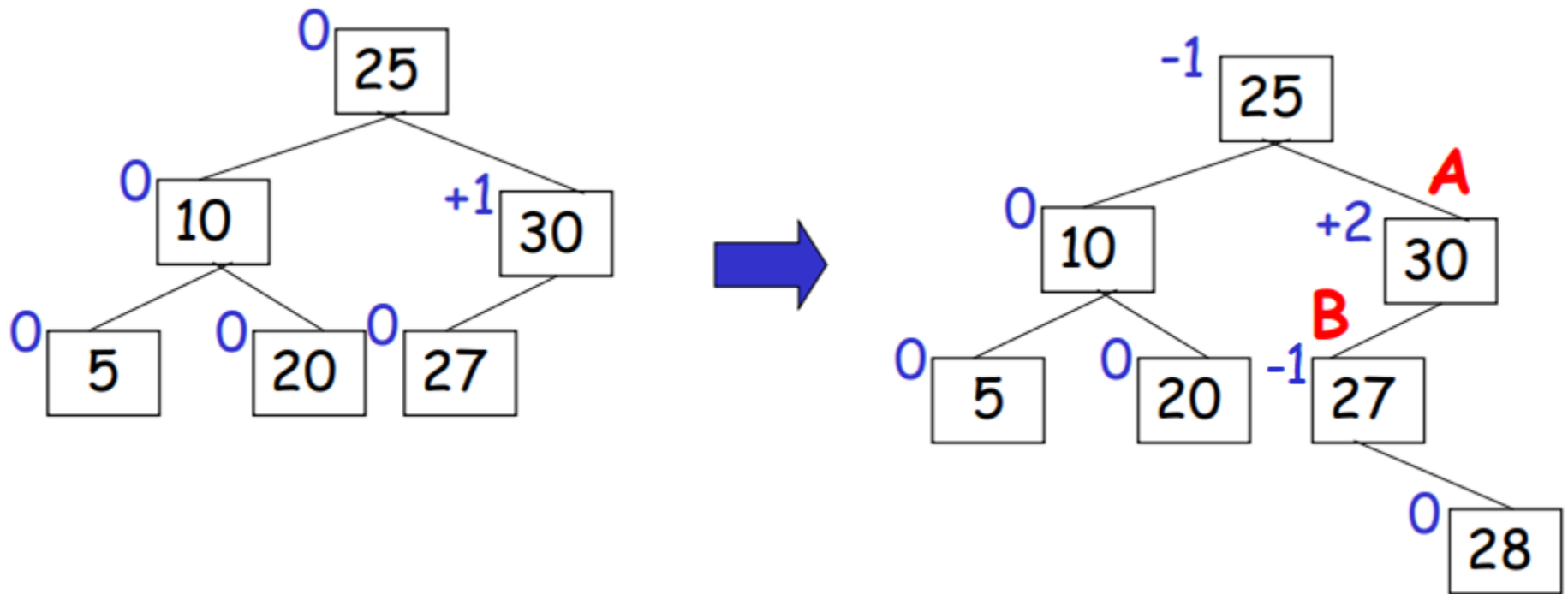
Rotação LR

- Passo a passo

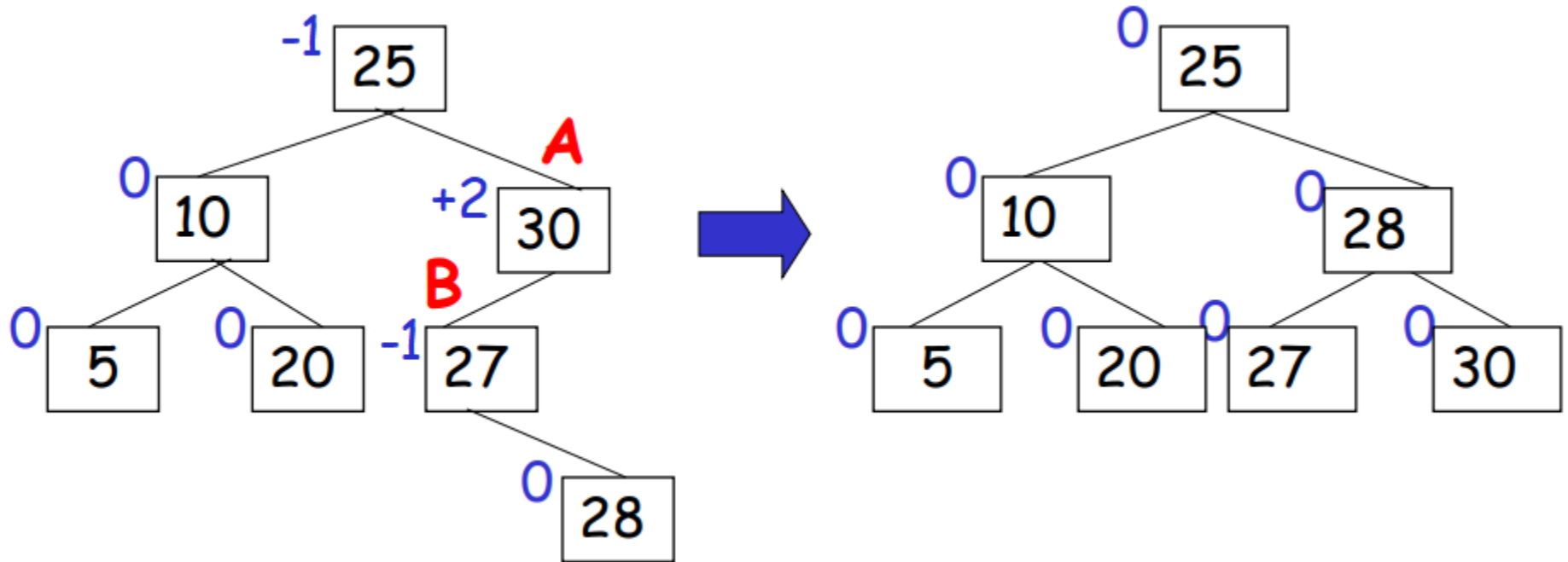


Árvore Balanceada

Rotação LR - Exemplo



Rotação LR - Exemplo

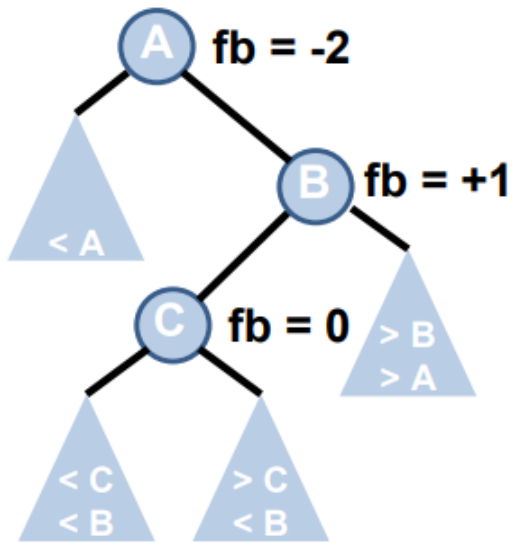


Rotação RL

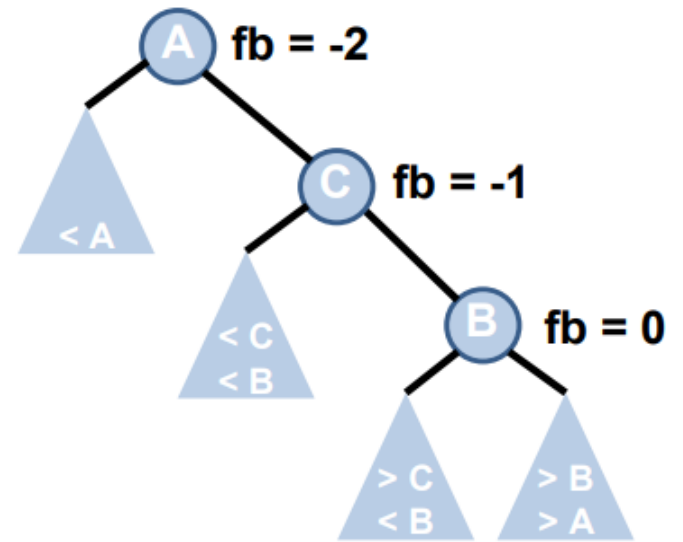
- Rotação RL ou rotação dupla à esquerda
- Um novo nó é inserido na subárvore da esquerda do filho direito de A
 - A é o nó desbalanceado
 - Um elemento para a direita e outro para a esquerda: **RIGHT LEFT**
- É necessário fazer uma rotação dupla, de modo que o nó C se torne o pai dos nós A (filho da esquerda) e B (filho da direita)
 - Rotação LL em B
 - Rotação RR em A

Rotação RL

- Exemplo

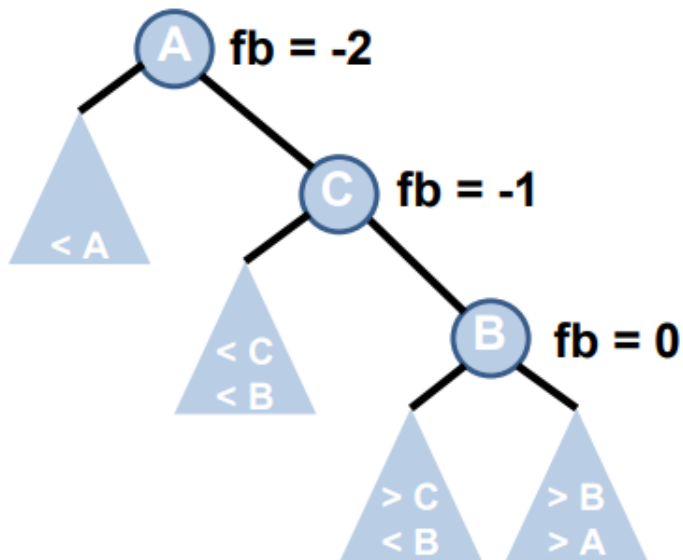



Rotação LL
em B

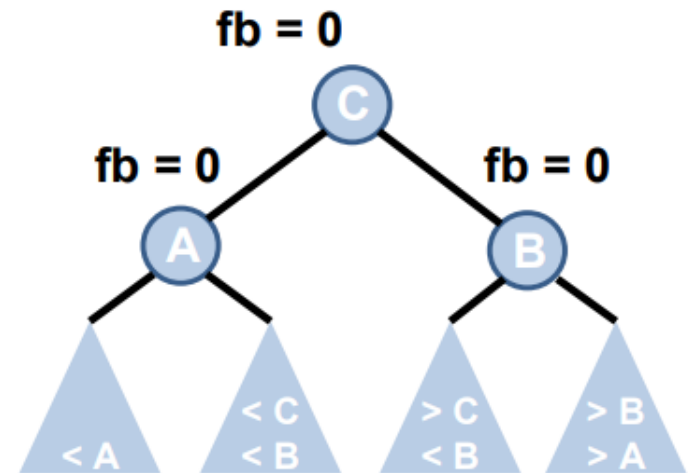


Rotação RL

- Exemplo

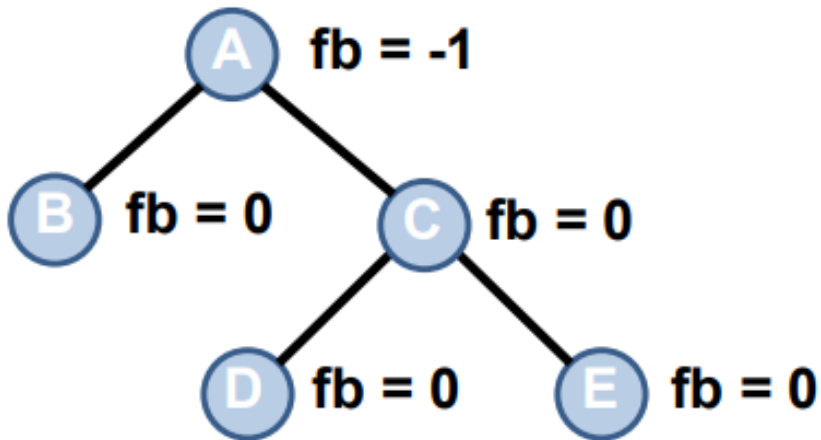



Rotação RR
em A



Rotação RL

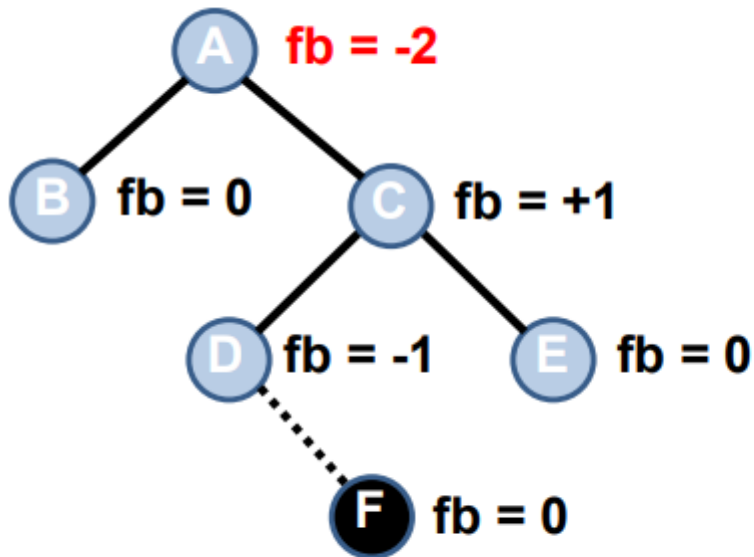
- Passo a passo



Árvore AVL e fator de balanceamento de cada nó

Rotação RL

- Passo a passo



Inserção do nó F na árvore

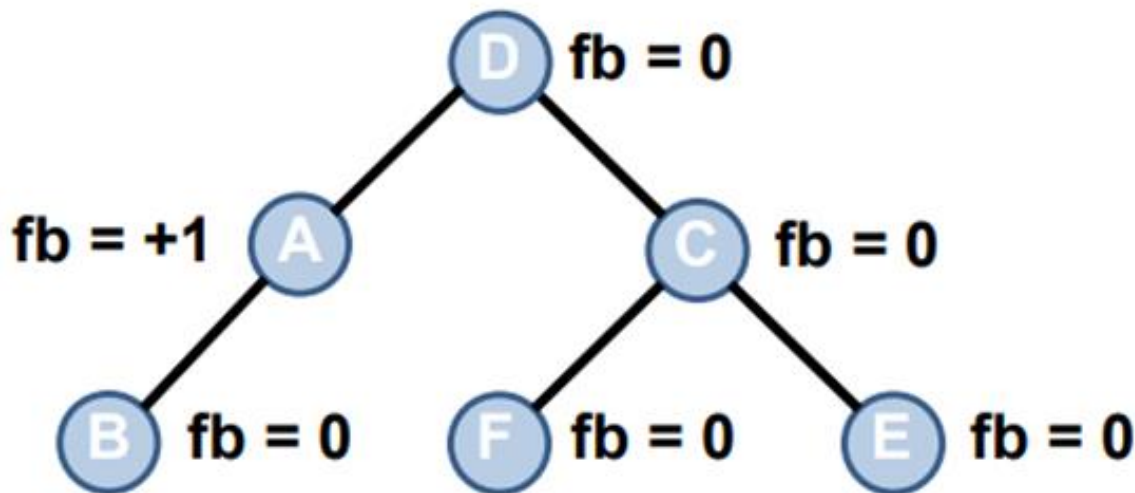
Árvore fica desbalanceada no nó A.

Aplicar Rotação RL no nó A.
Isso equivale a:

- Aplicar a Rotação LL no nó C
- Aplicar a Rotação RR no nó A

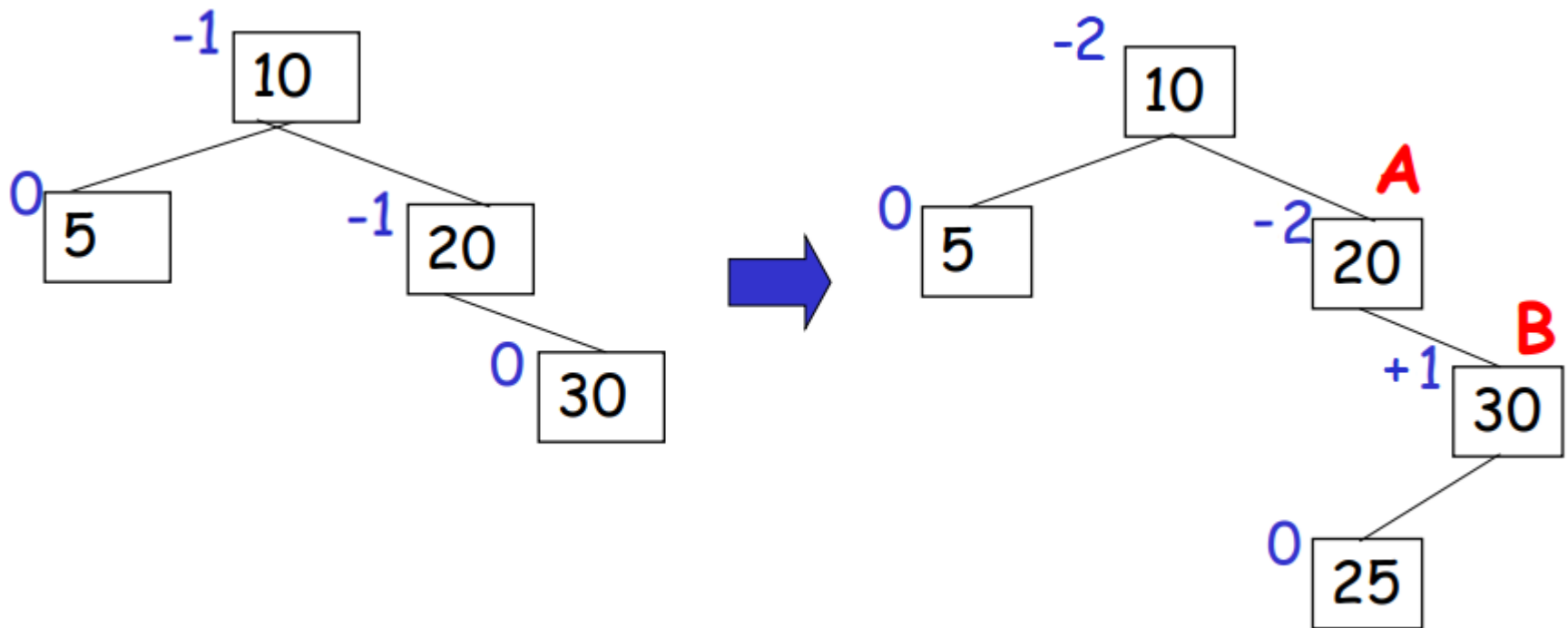
Rotação RL

- Passo a passo

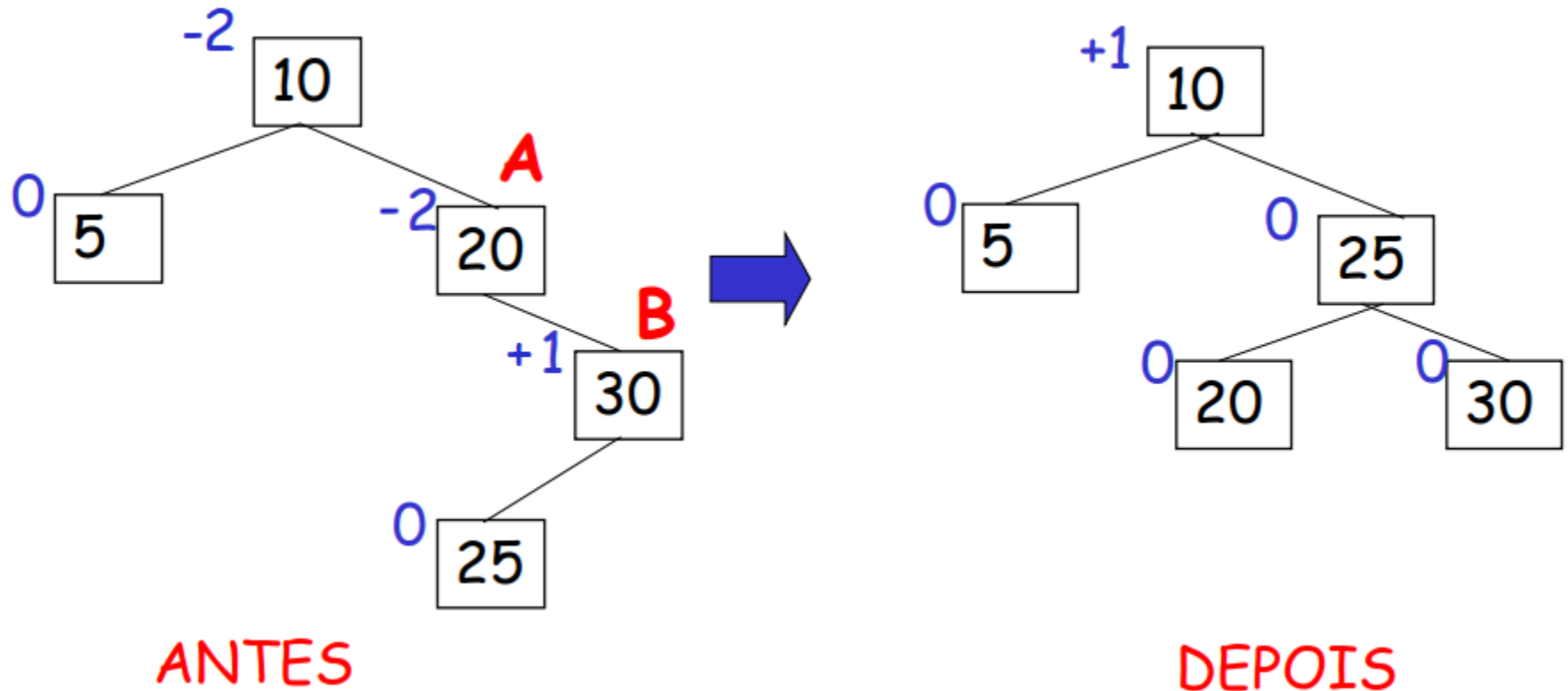


Árvore Balanceada

Rotação RL - Exemplo



Rotação RL - Exemplo



Quando usar cada rotação?

- Sinais iguais: rotação simples
- Sinal positivo: rotação à direita (LL)
- Sinal negativo: rotação à esquerda (RR)

Fator de Balanceamento de A	Fator de Balanceamento de B	Posições dos nós B e C em relação ao nó A	Rotação
+2	+1	B é filho à esquerda de A C é filho à esquerda de B	LL
-2	-1	B é filho à direita de A C é filho à direita de B	RR
+2	-1	B é filho à esquerda de A C é filho à direita de B	LR
-2	+1	B é filho à de direita A C é filho à esquerda de B	RL

Quando usar cada rotação?

- Sinais diferentes: rotação dupla
- A positivo: rotação dupla a direita (LR)
- A negativo: rotação dupla a esquerda (RL)

Fator de Balanceamento de A	Fator de Balanceamento de B	Posições dos nós B e C em relação ao nó A	Rotação
+2	+1	B é filho à esquerda de A C é filho à esquerda de B	LL
-2	-1	B é filho à direita de A C é filho à direita de B	RR
+2	-1	B é filho à esquerda de A C é filho à direita de B	LR
-2	+1	B é filho à de direita A C é filho à esquerda de B	RL

Exemplo de sucessivas inserções

Inserção - Maio

Depois da inserção

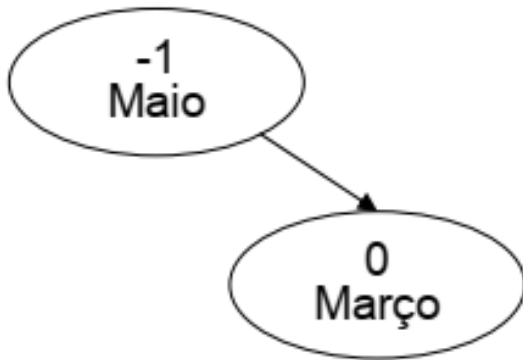


Depois do rebalanceamento

*Sem necessidade
de rebalanceamento*

Inserção - Março

Depois da inserção

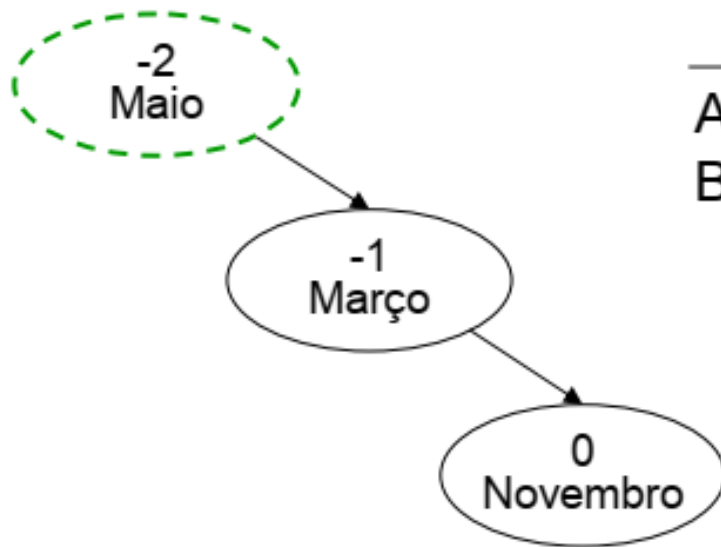


Depois do rebalanceamento

*Sem necessidade
de rebalanceamento*

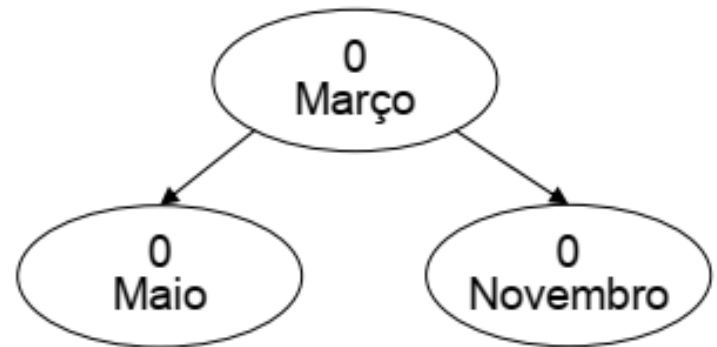
Inserção - Novembro

Depois da inserção



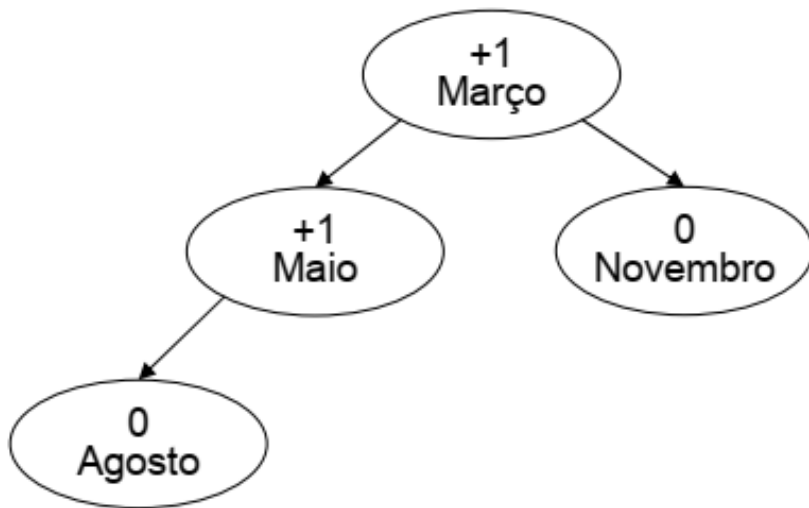
RR
A = -2
B = -1

Depois do rebalanceamento



Inserção - Agosto

Depois da inserção

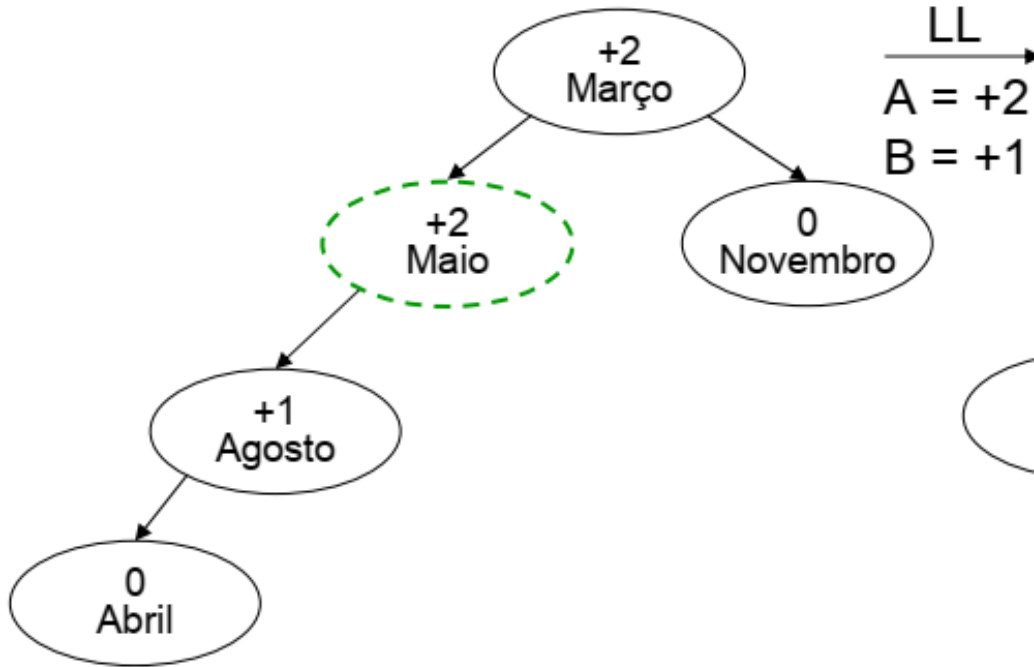


Depois do rebalanceamento

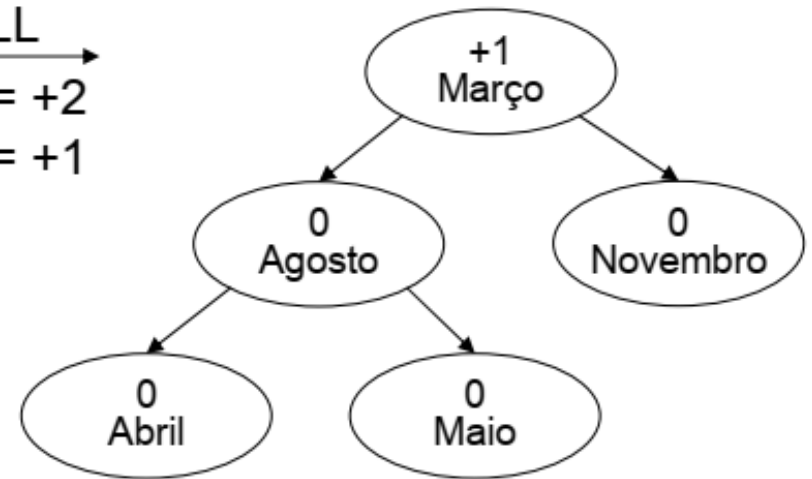
*Sem necessidade
de rebalanceamento*

Inserção - Abril

Depois da inserção

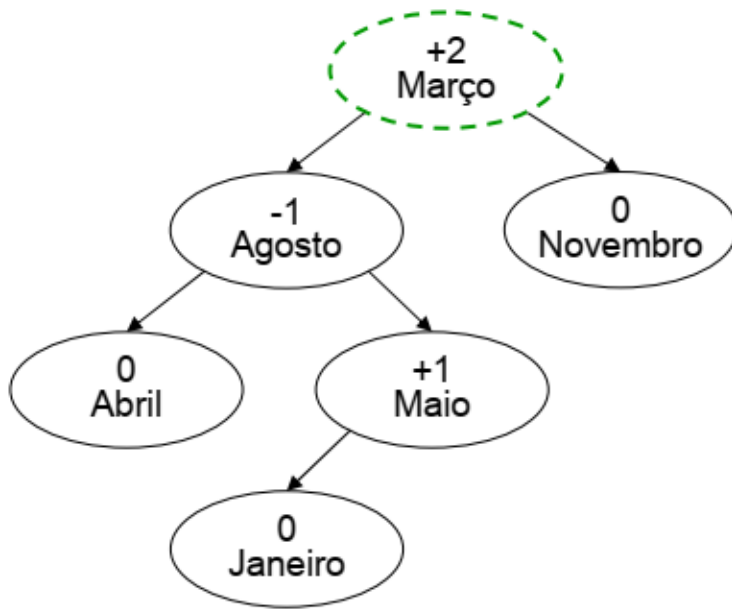


Depois do rebalanceamento



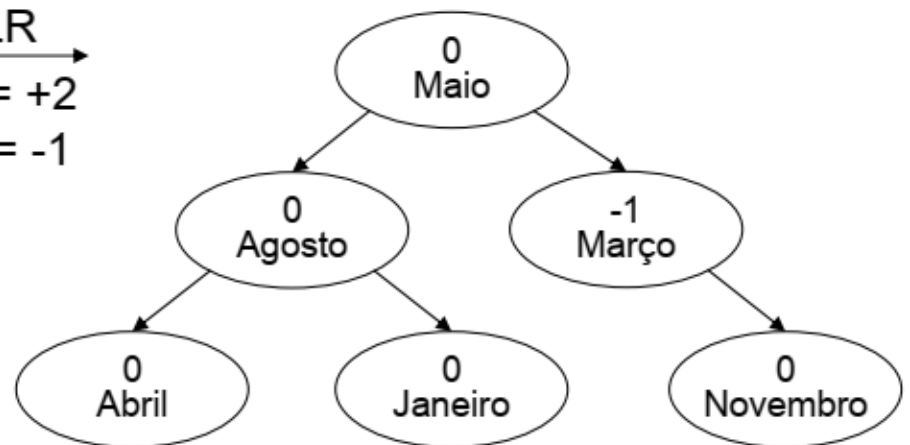
Inserção - Janeiro

Depois da inserção



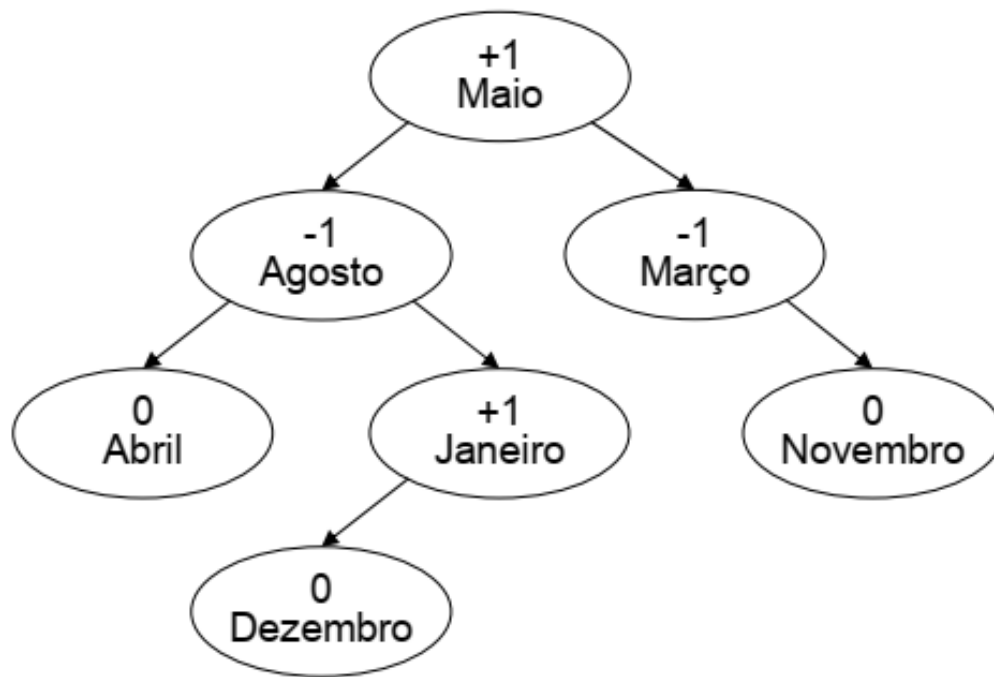
Depois do rebalanceamento

LR
A = +2
B = -1



Inserção - Dezembro

Depois da inserção

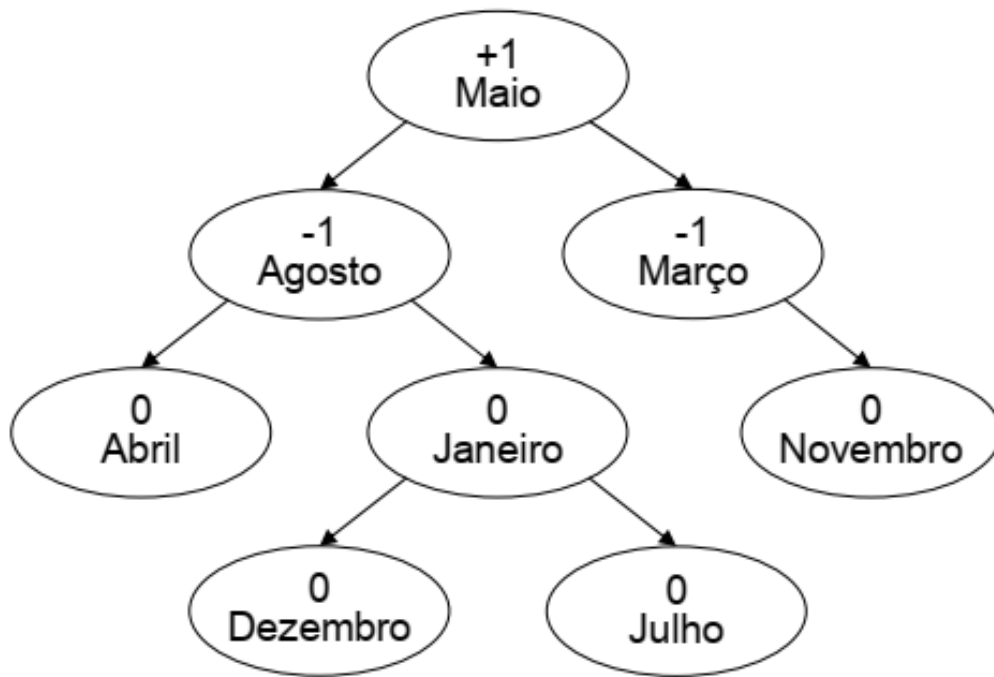


Depois do rebalanceamento

*Sem necessidade
de rebalanceamento*

Inserção - Julho

Depois da inserção

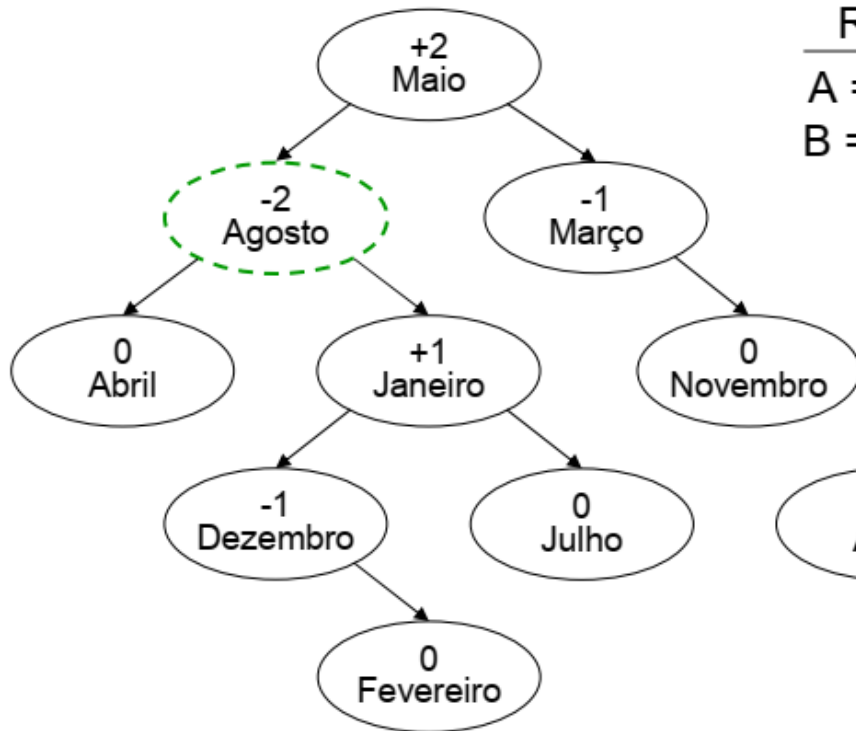


Depois do rebalanceamento

*Sem necessidade
de rebalanceamento*

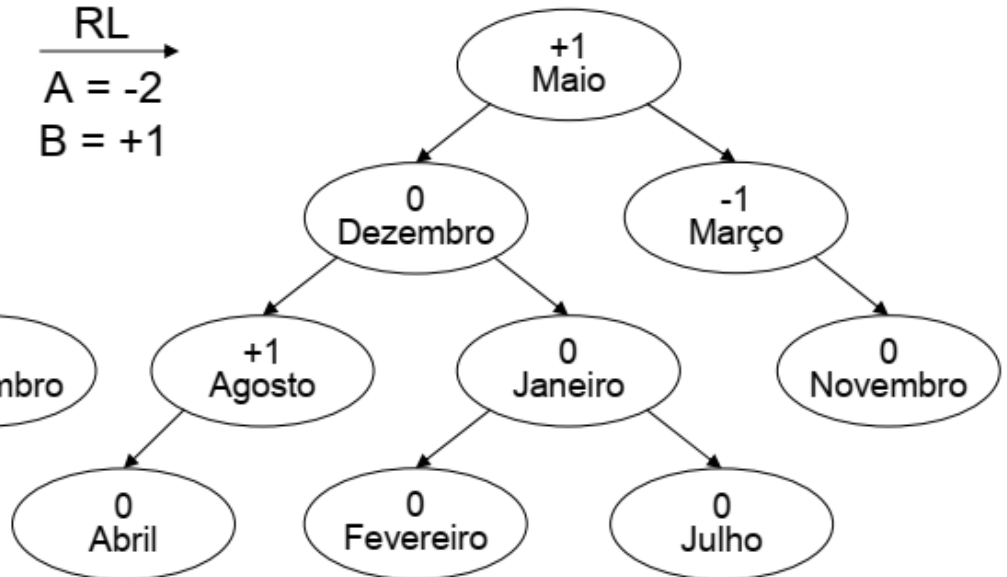
Inserção - Fevereiro

Depois da inserção



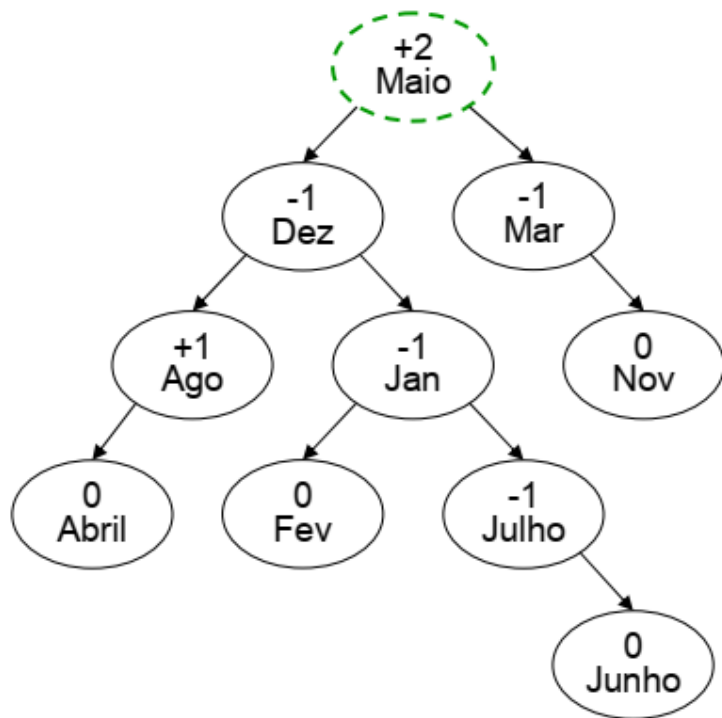
RL
A = -2
B = +1

Depois do rebalanceamento

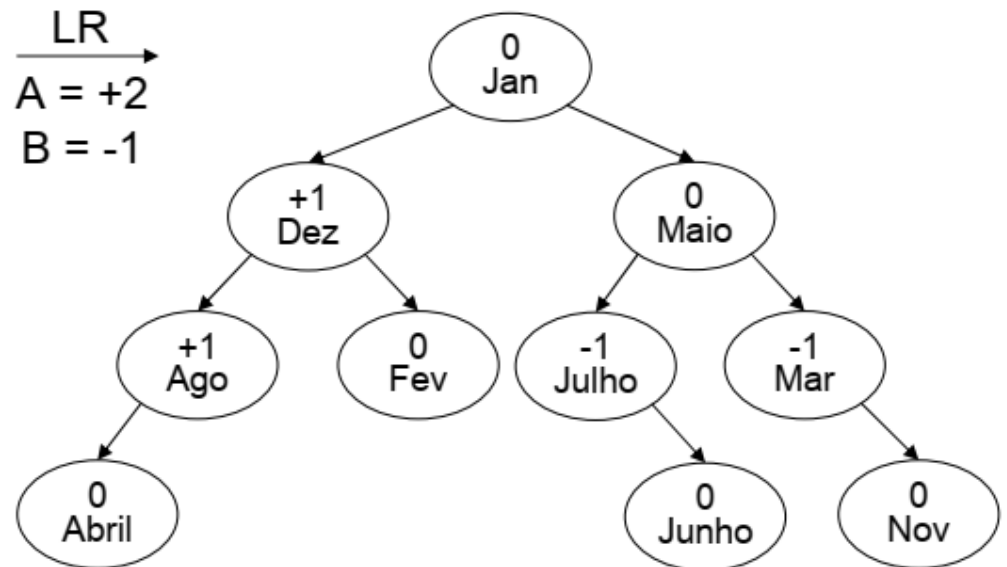


Inserção - Junho

Depois da inserção

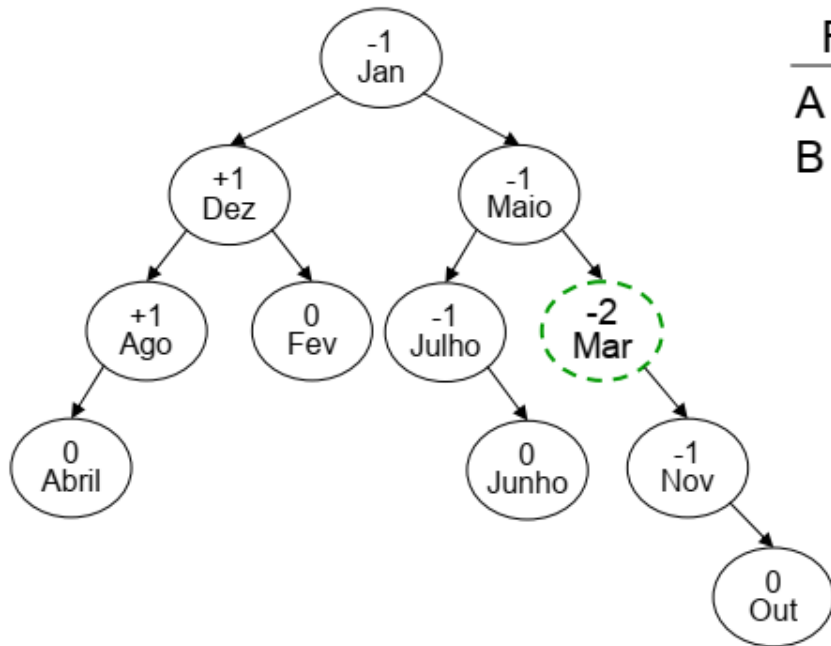


Depois do rebalanceamento



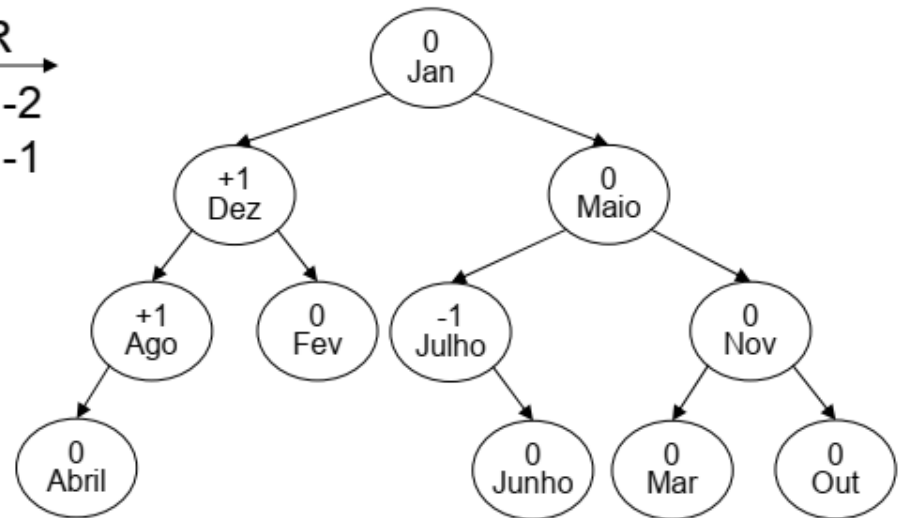
Inserção - Outubro

Depois da inserção



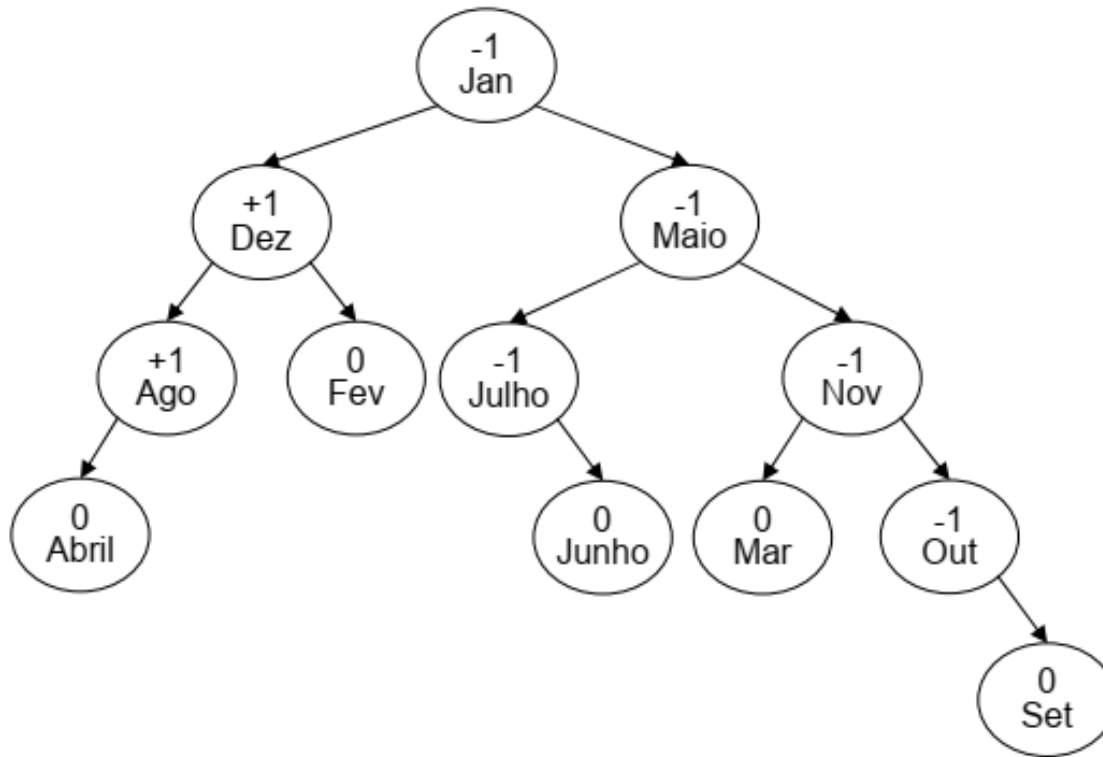
RR
A = -2
B = -1

Depois do rebalanceamento



Inserção - Setembro

Depois da inserção



Depois do rebalanceamento

*Sem necessidade
de rebalanceamento*

Árvore AVL: Remoção

- Como na inserção, temos que percorremos um conjunto de nós da árvore até chegar ao nó que será removido
- Existem 3 tipos de remoção
 - Nó folha (sem filhos)
 - Nó com 1 filho
 - Nó com 2 filhos*

*Substituir pelo nó mais à direita da subárvore esquerda (maior elemento da subárvore esquerda)

Árvore AVL: Remoção

Uma vez removido o nó:

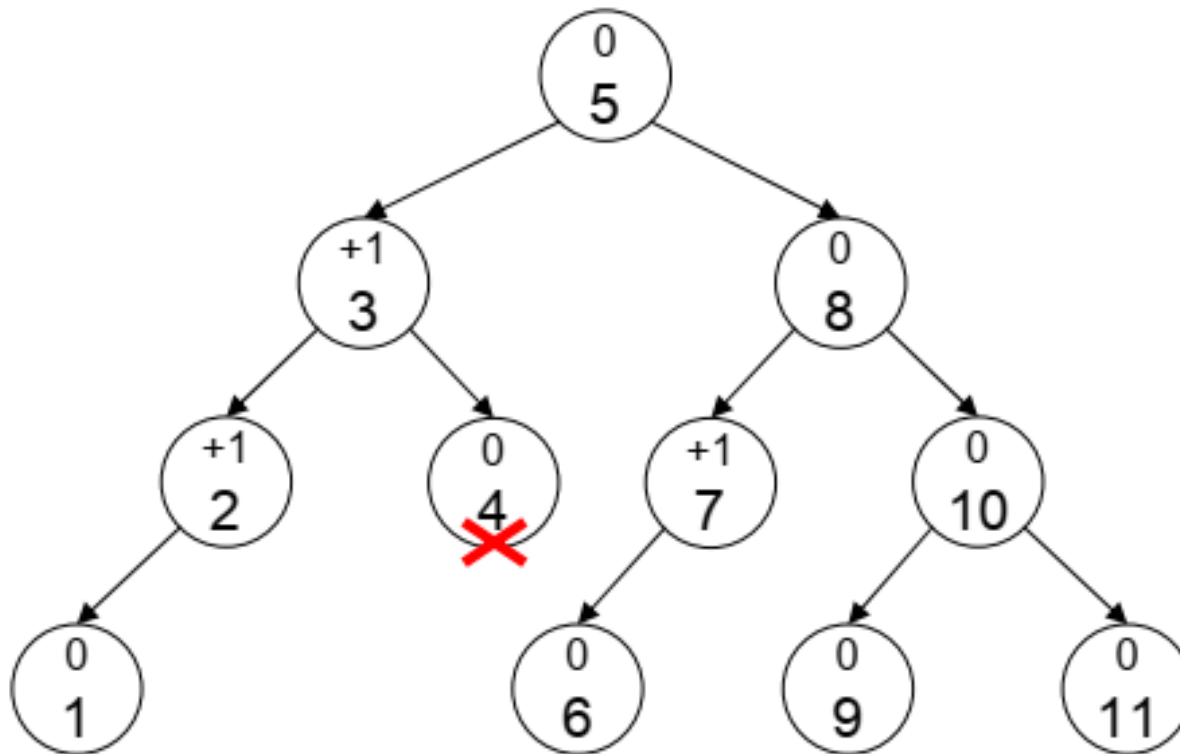
- Devemos voltar pelo caminho percorrido e calcular o fator de balanceamento de cada um dos nós visitados
- Aplicar a rotação necessária para restabelecer o balanceamento da árvore se o fator de balanceamento for +2 ou -2
 - Remover um nó da subárvore direita equivale a inserir um nó na subárvore esquerda

A large teal shape that starts as a rectangle at the top and curves downwards towards the bottom right, creating a modern, abstract background element.

Exemplo de sucessivas remoções

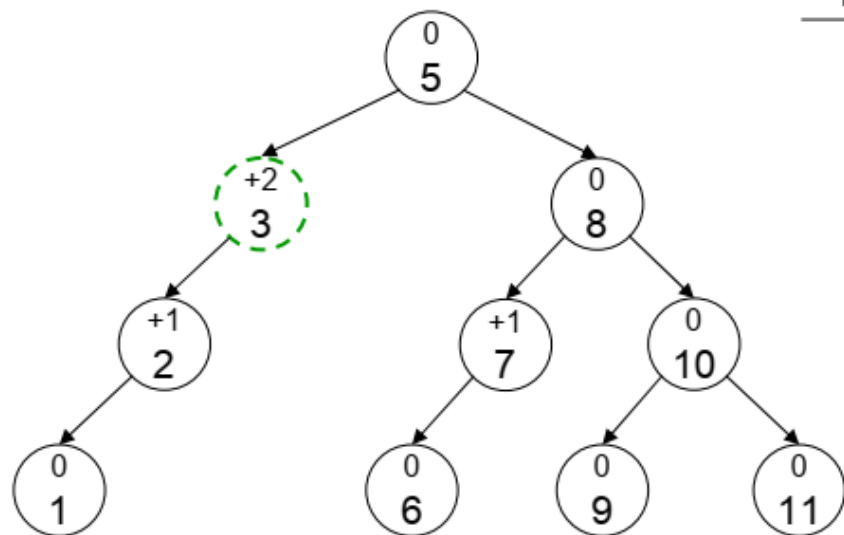
Remoção - 4

Antes da remoção



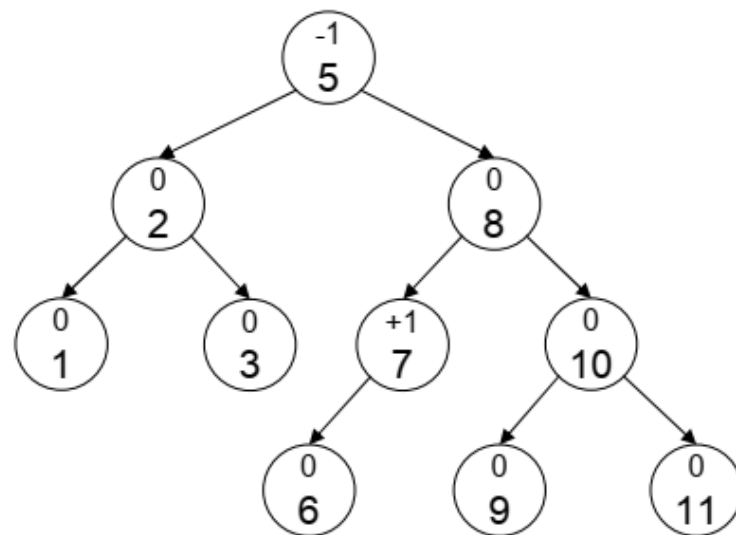
Remoção - 4

Depois da remoção



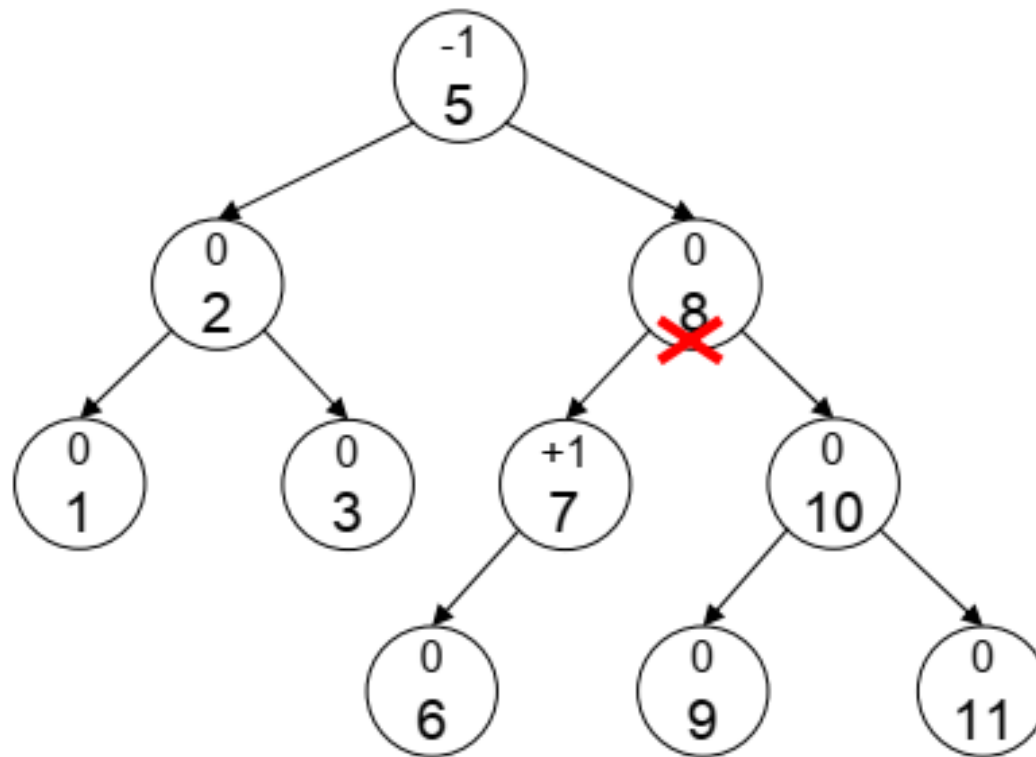
LL →

Depois do rebalanceamento



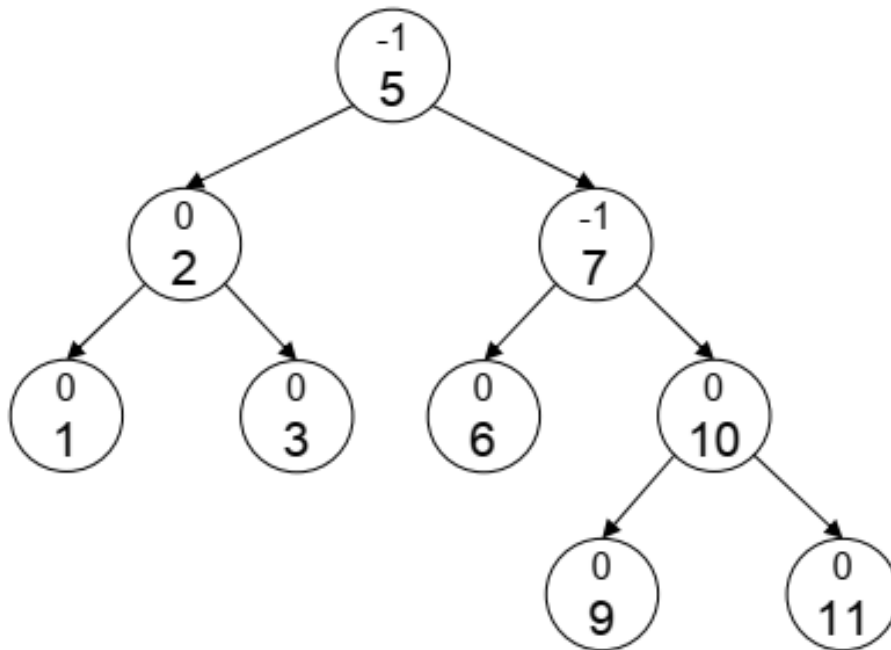
Remoção - 8

Antes da remoção



Remoção - 8

Depois da remoção



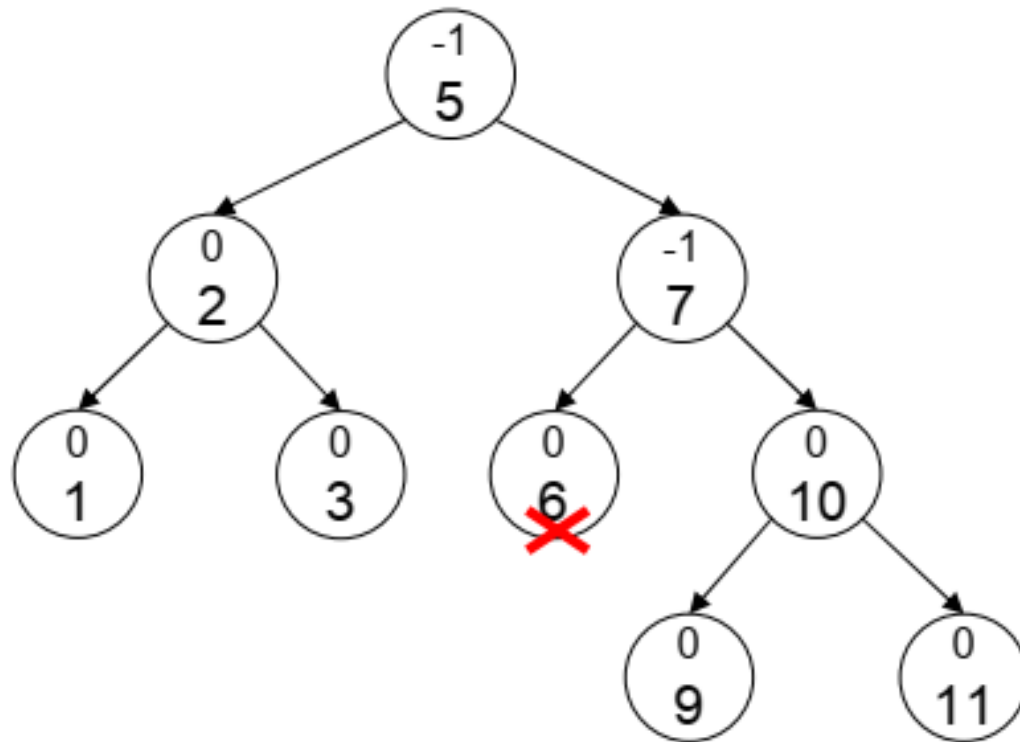
Depois do rebalanceamento

*Sem necessidade
de rebalanceamento*

Obs.: Foi utilizado o maior elemento da subárvore esquerda do nó sendo removido

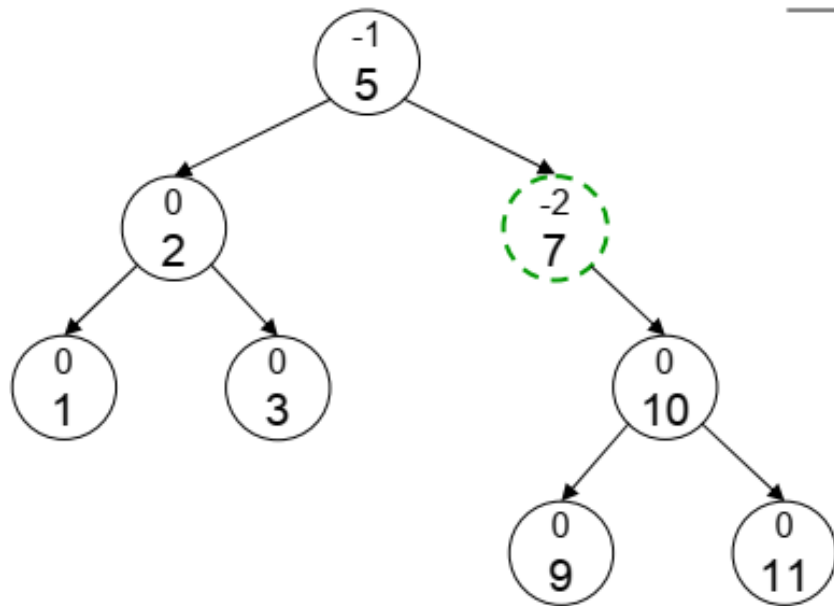
Remoção - 6

Antes da remoção



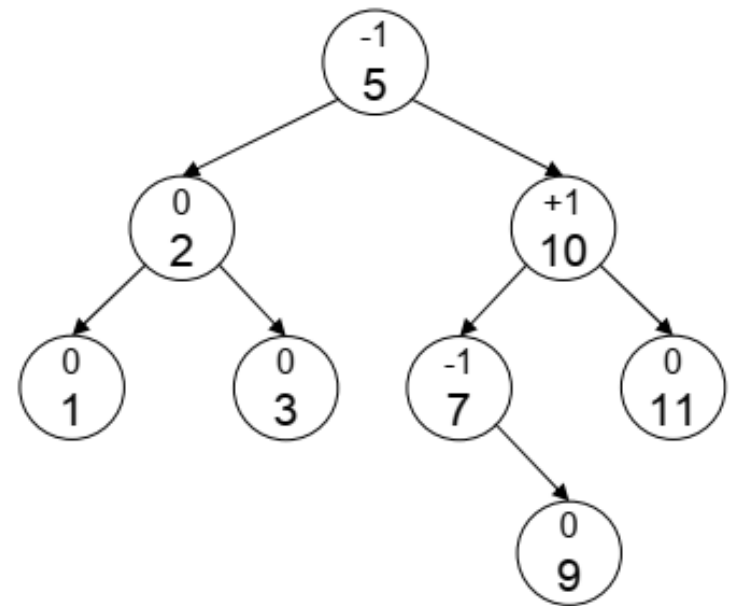
Remoção - 6

Depois da remoção



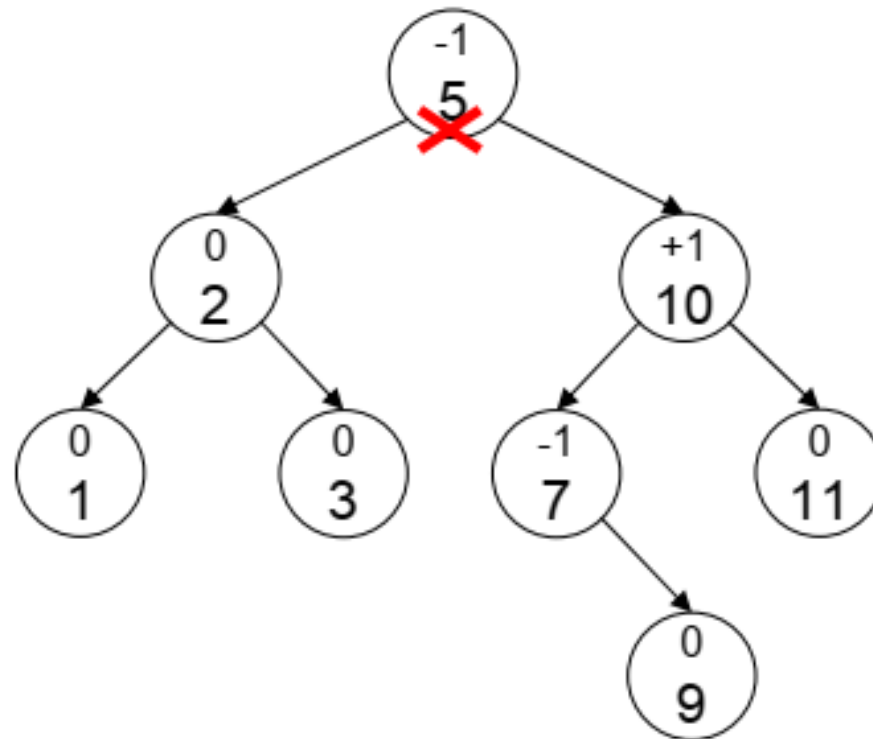
RR →

Depois do rebalanceamento



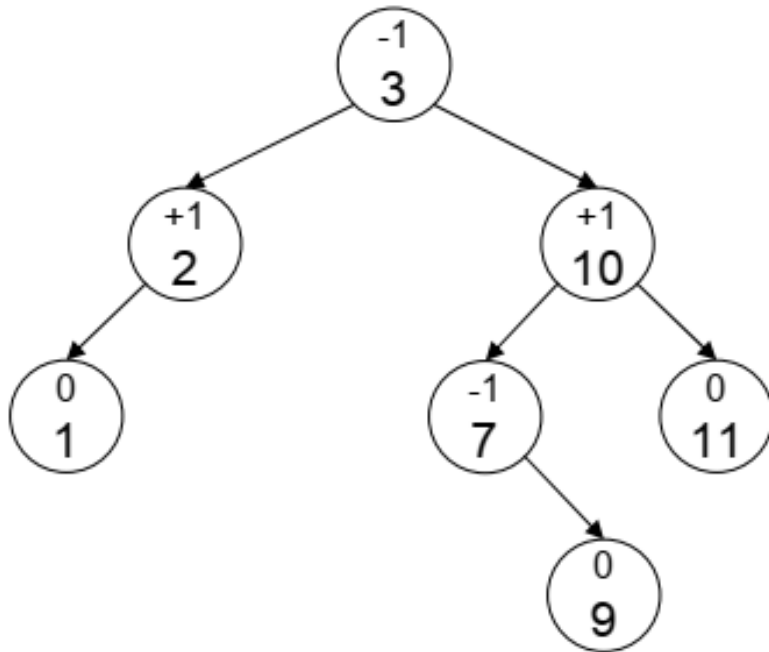
Remoção - 5

Antes da remoção



Remoção - 5

Depois da remoção

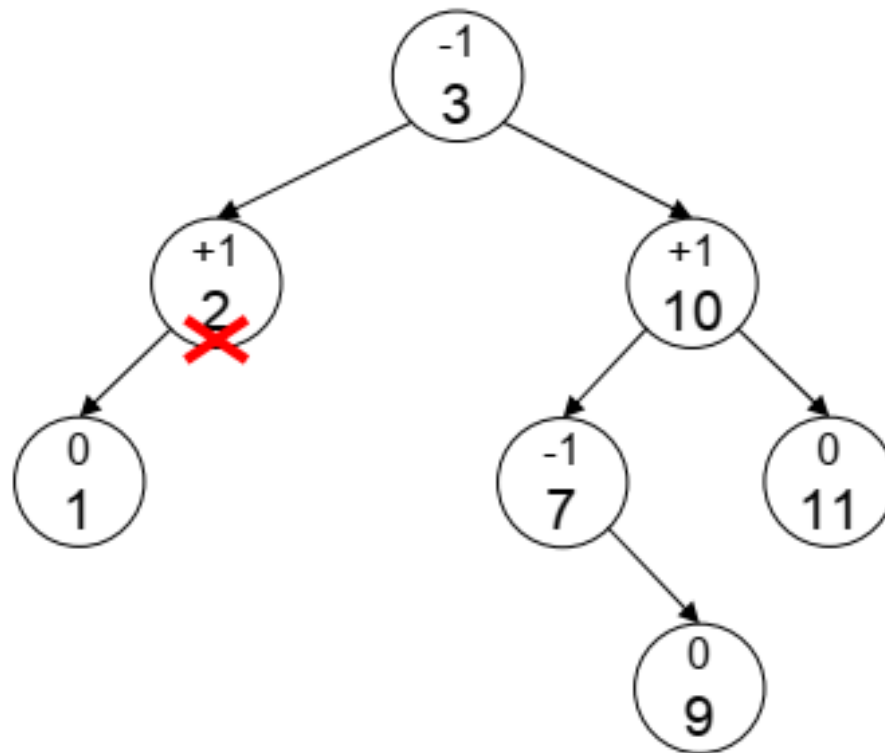


Depois do rebalanceamento

*Sem necessidade
de rebalanceamento*

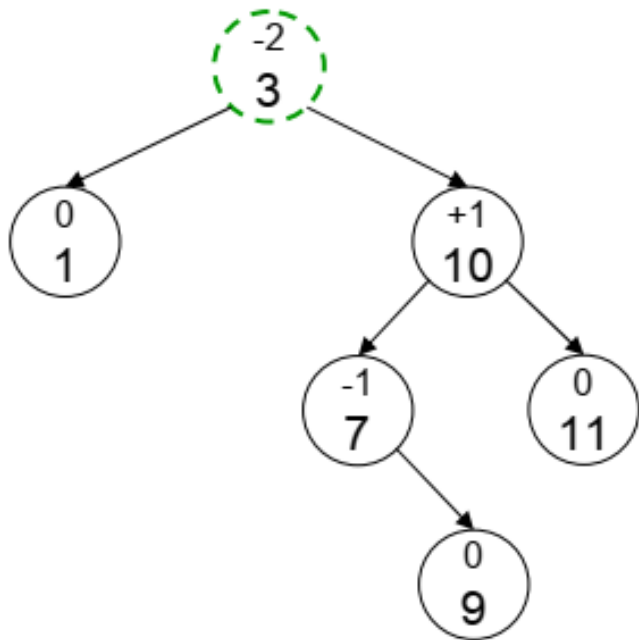
Remoção - 2

Antes da remoção



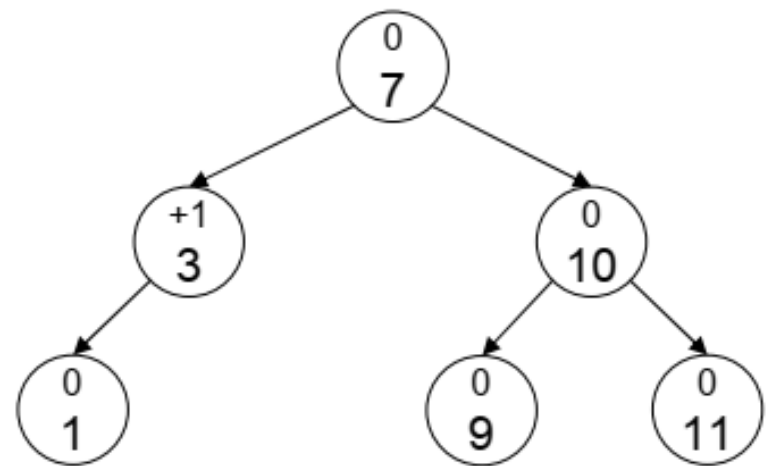
Remoção - 2

Depois da remoção



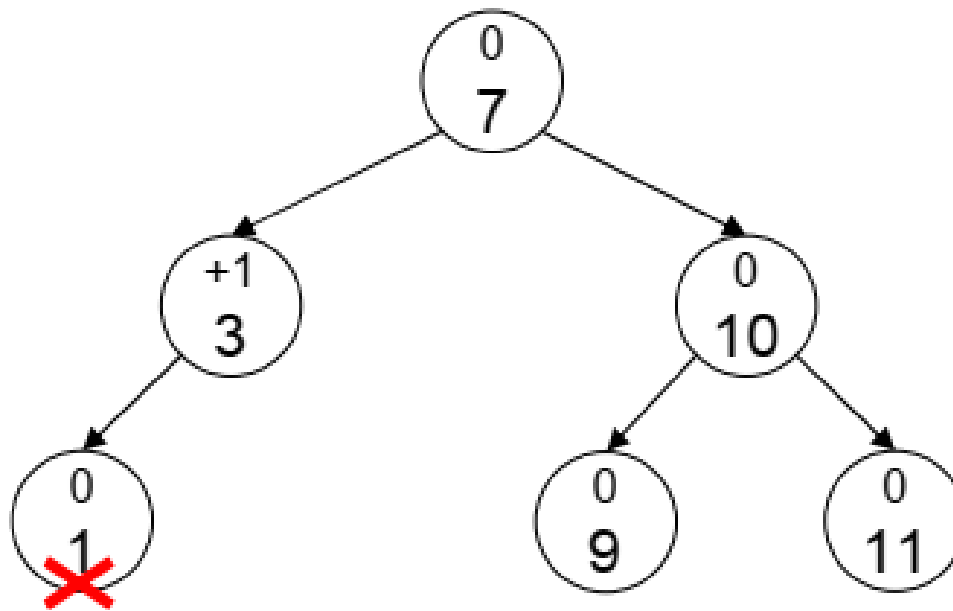
RL →

Depois do rebalanceamento



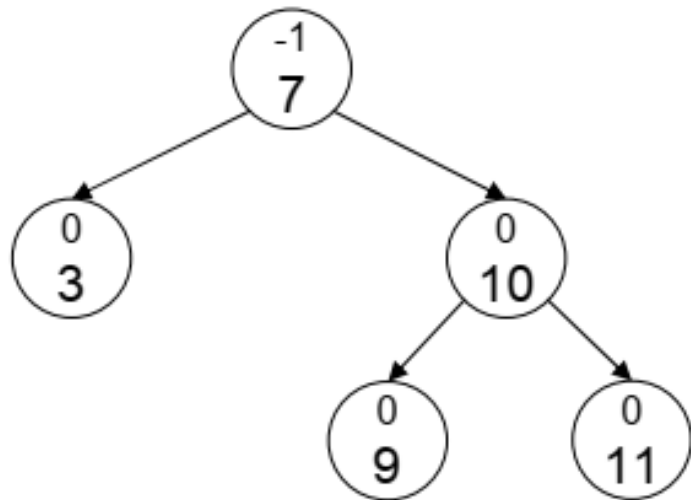
Remoção - 1

Antes da remoção



Remoção - 1

Depois da remoção

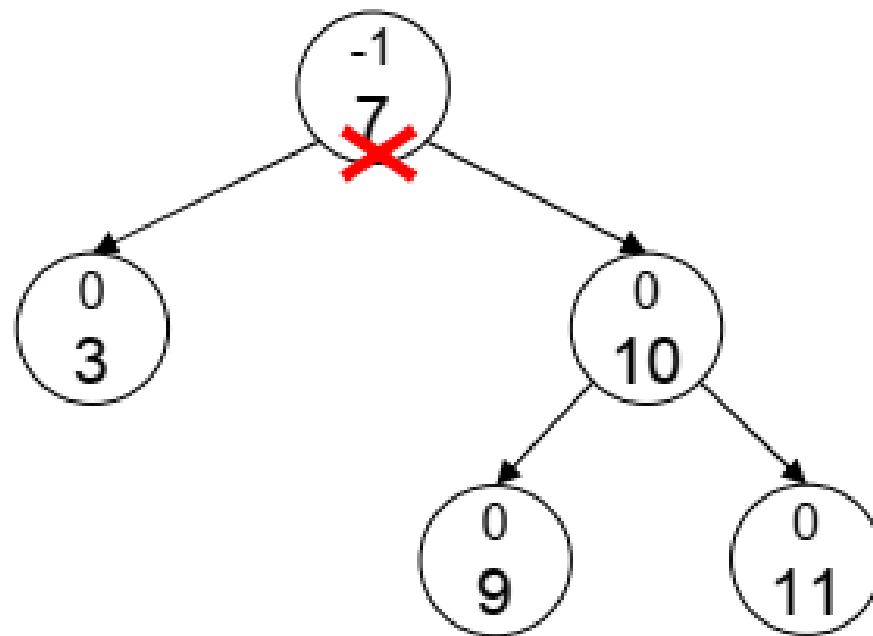


Depois do rebalanceamento

*Sem necessidade
de rebalanceamento*

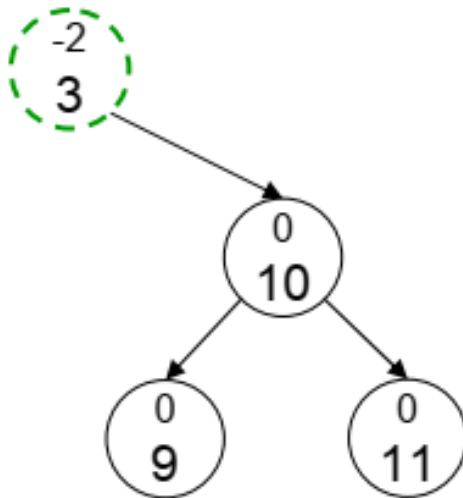
Remoção - 7

Antes da remoção



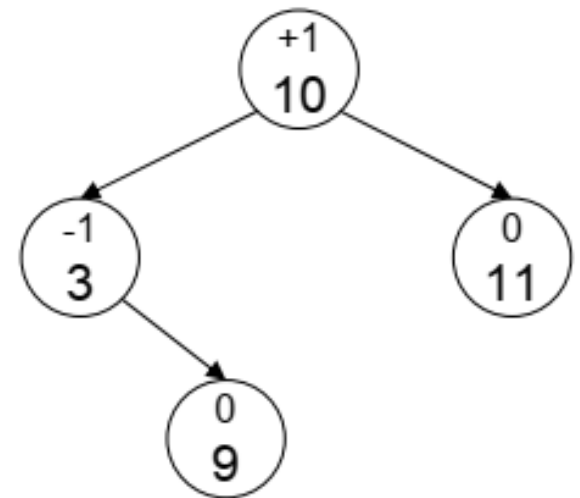
Remoção - 7

Depois da remoção



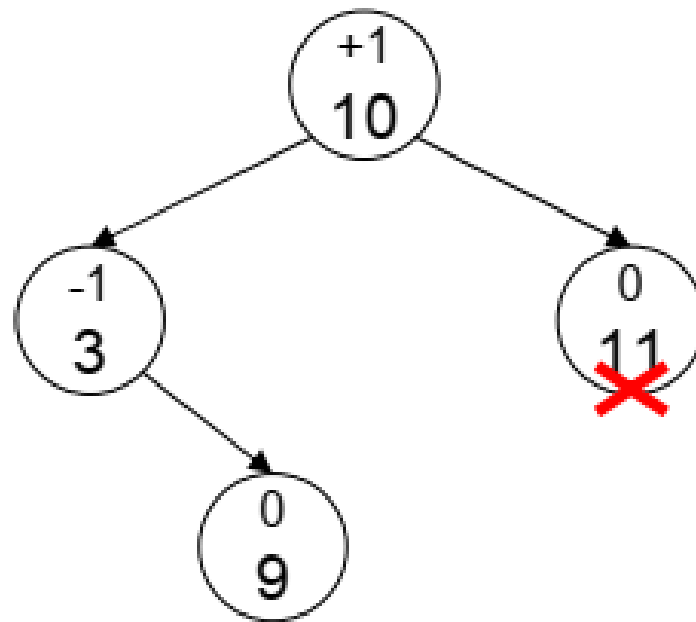
RR →

Depois do rebalanceamento



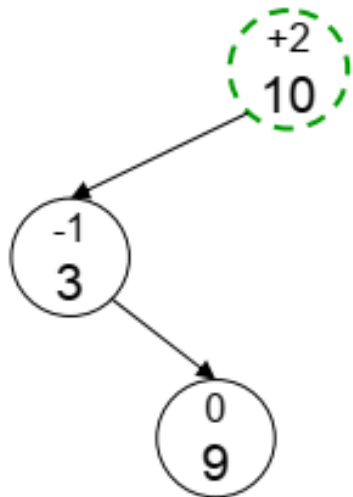
Remoção - 11

Antes da remoção



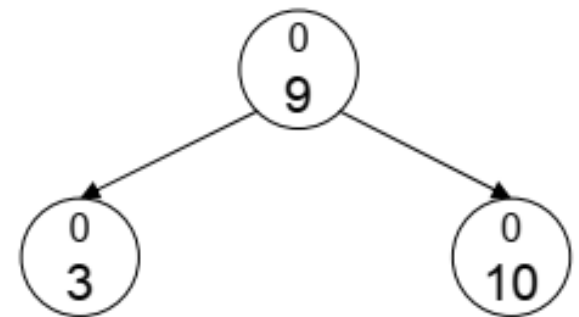
Remoção - 11

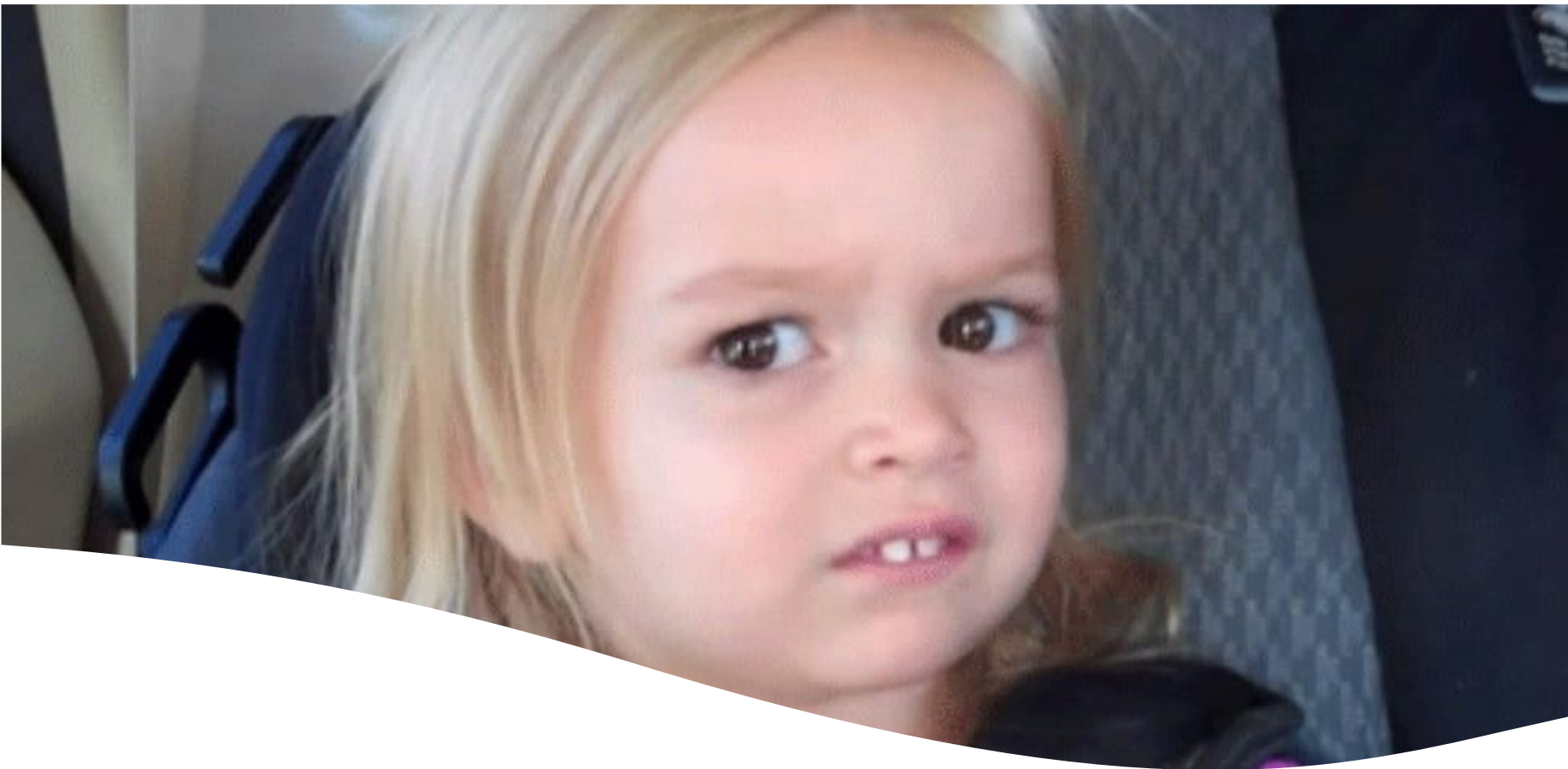
Depois da remoção



LR →

Depois do rebalanceamento





Atividade

Exercícios



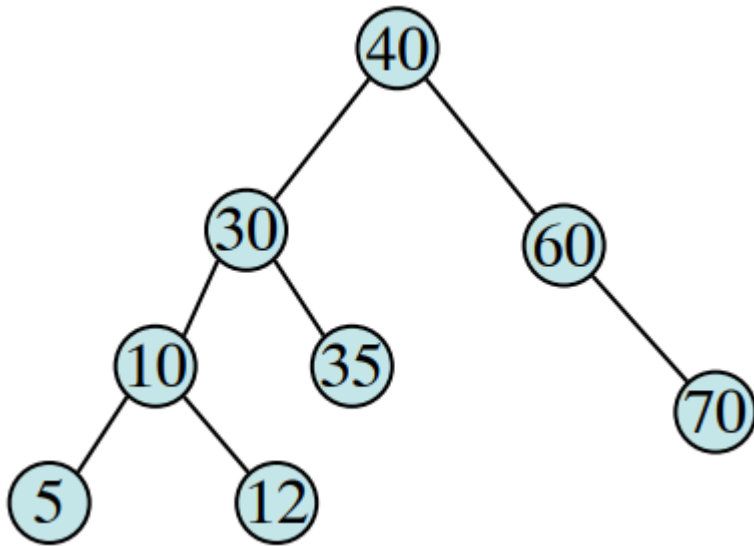
uniderp

Exercício 1

- Inserir os elementos a seguir em uma árvore AVL, mostrando a árvore em cada etapa: 4, 5, 7, 2, 1, 3, 6

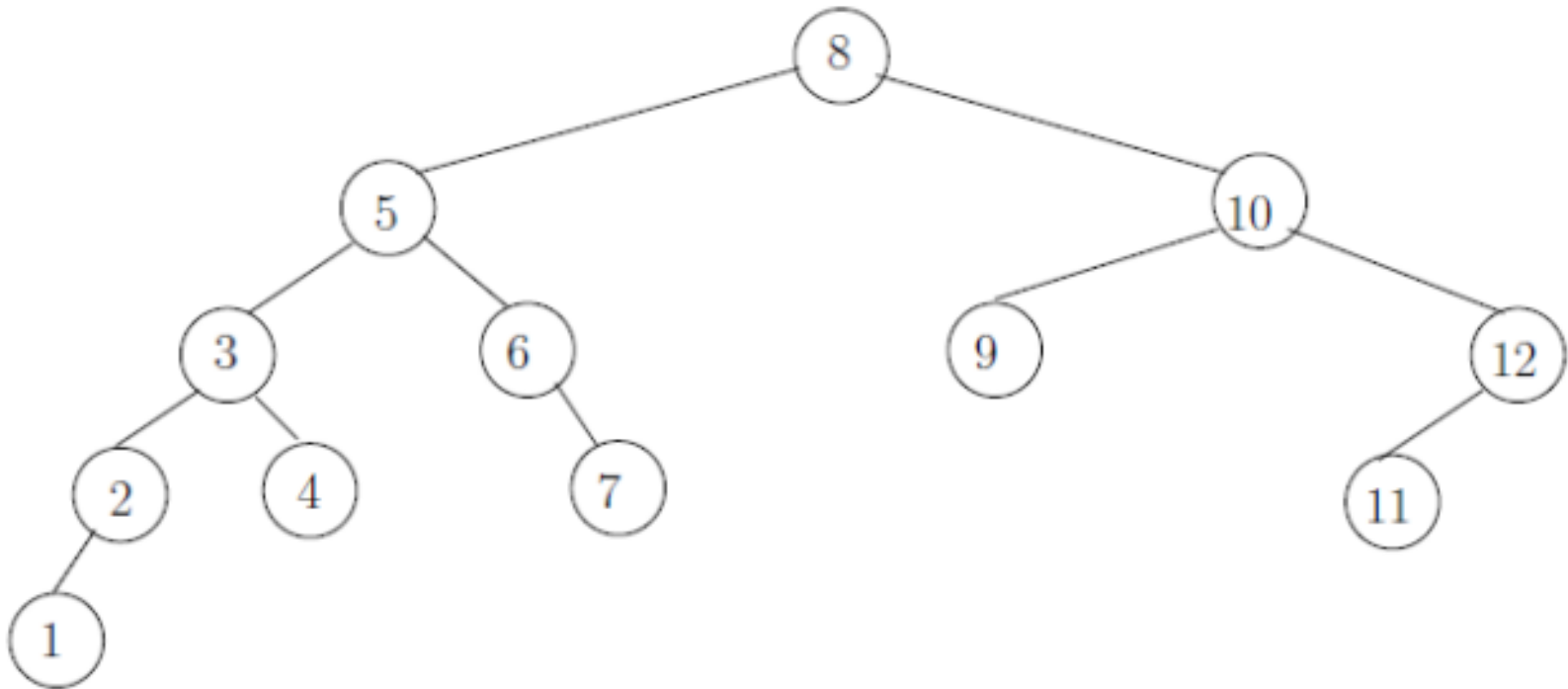
Exercício 2

- Inserir na árvore AVL abaixo os seguintes elementos: 3, 33, 11 e 9



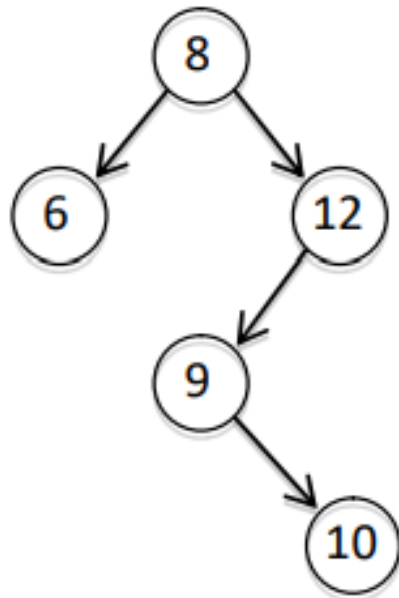
Exercício 3

- Remova da seguinte árvore AVL a seguinte sequência de valores: 8, 10, 3, 1, 7.

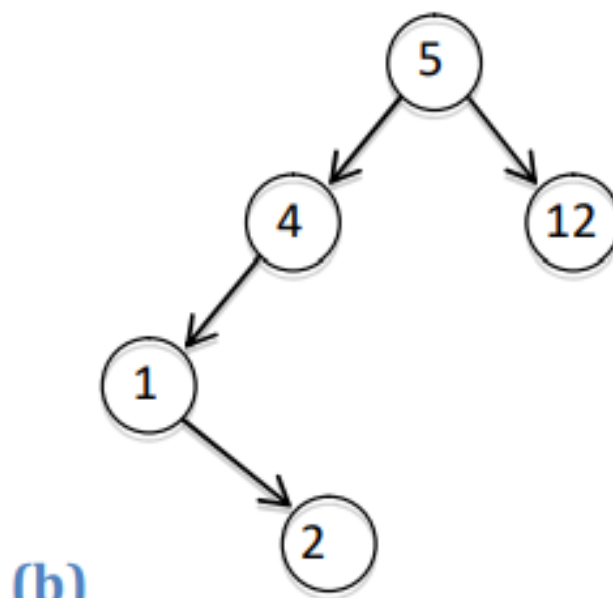


Exercício 4

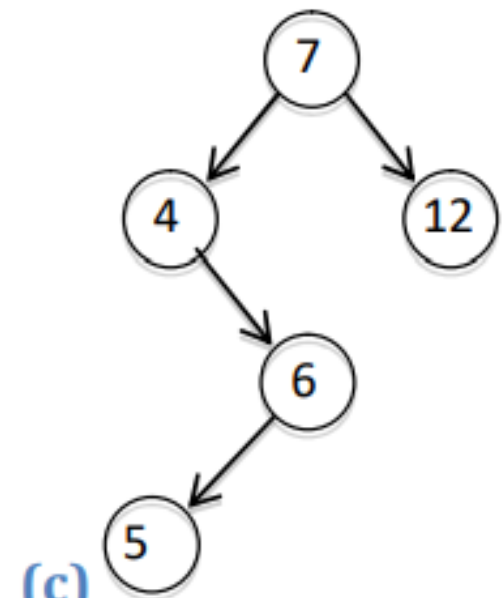
- Analise os casos abaixo considerando uma árvore AVL e identifique o tipo de transformação de balanceamento que deve ser efetuado. Mostre o resultado após o balanceamento.



(a)

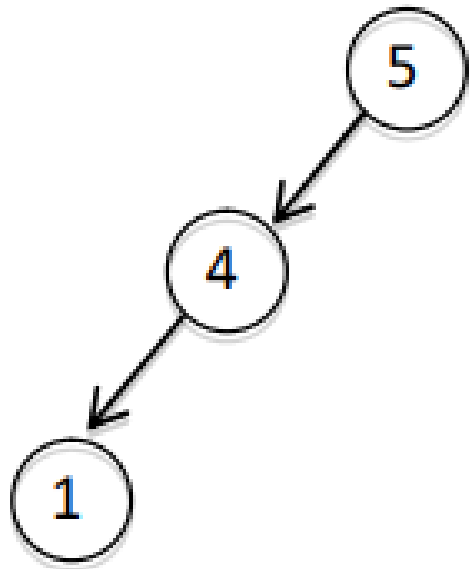


(b)

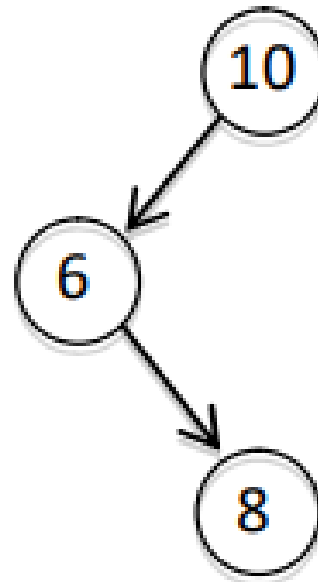


(c)

Exercício 4 (continuação)



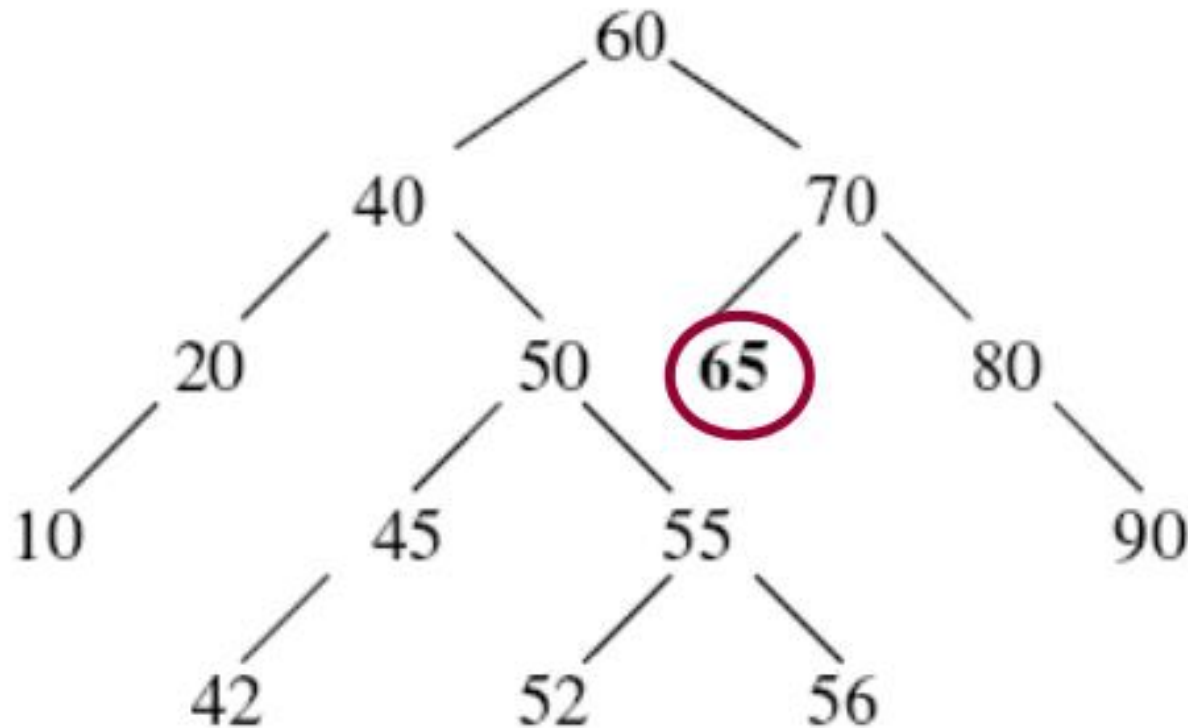
(d)



(e)

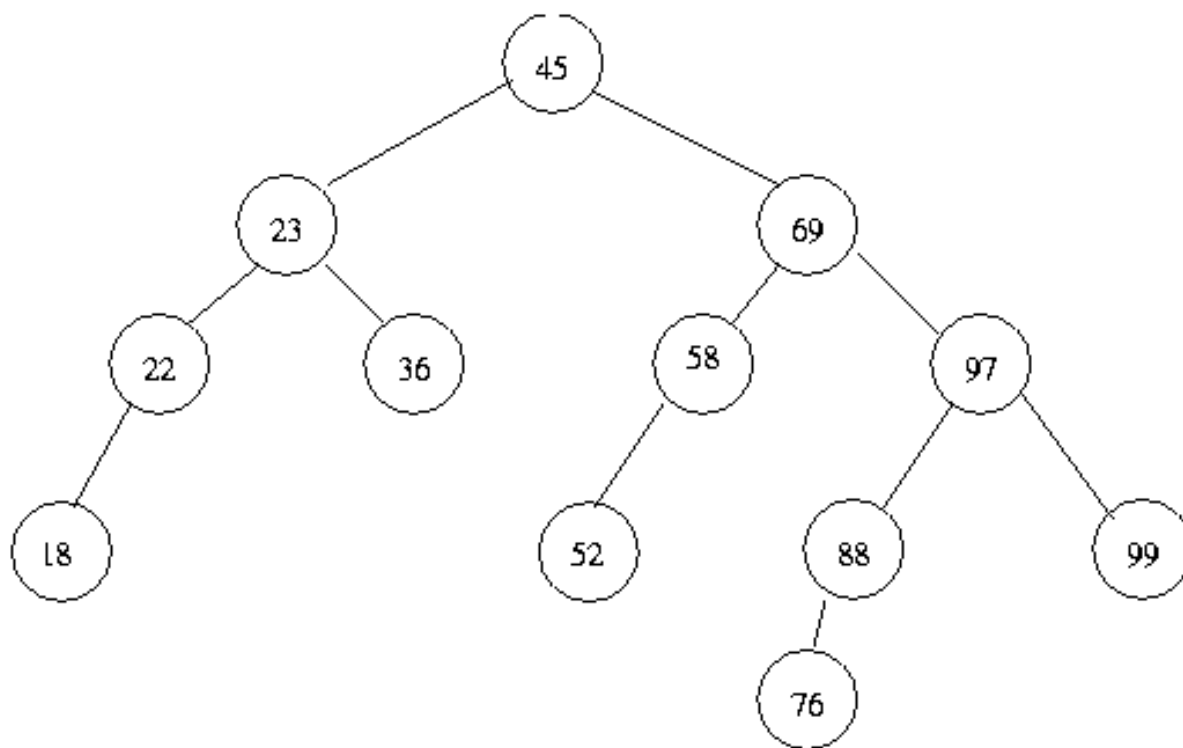
Exercício 5

- Remova da seguinte árvore AVL o elemento 65.



Exercício 6

- Nesta questão, você deverá executar a lista de operações na árvore AVL mostrada baixo. Desenhe a árvore resultante da operação correspondente.



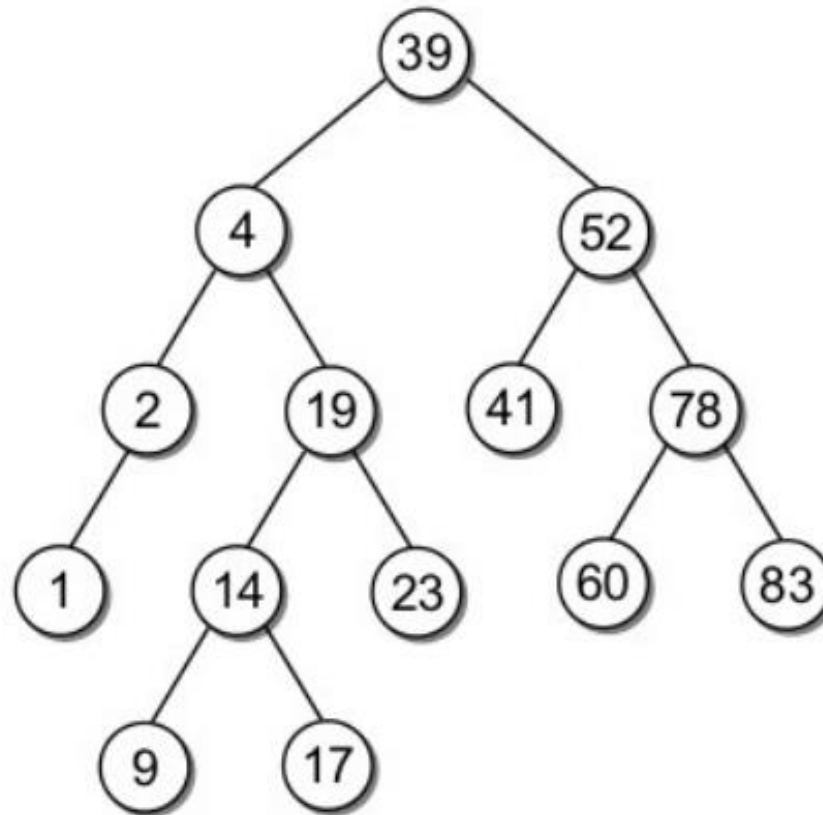
- Insira 105
- Insira 20
- Remova 99
- Remova 36

Exercício 7

- Dada uma AVL que é uma folha com a chave 50
- (a) Insira os elementos {1, 64, 12, 18, 66, 38, 95, 58, 59, 70, 68, 39, 62, 7, 60, 43, 16, 67, 34, 35} nesta árvore, indicando as rotações necessárias;
- (b) A ordem em que os elementos são inseridos numa árvore AVL não importa, pois independente da ordem, sempre resulta na mesma árvore. Esta afirmação está correta?
- (c) Remova os elementos {50, 95, 70, 60, 35} desta árvore, explicitando as rotações

Exercício 8

- Considere a árvore AVL abaixo e mostre o resultado dessa árvore após remover as chaves 1, 78 e 41.



Encerramento



DÚVIDAS, CRÍTICAS E SUGESTÕES, ENVIAR PARA:

noiza@anhanguera.com



uniderp