



AED2

Algoritmos e Estruturas de Dados II

Aula 10 – Árvores Adelson-Velskii e Landis

Prof. Aléssio Miranda Júnior

alessio@timoteo.cefetmg.br

2Q-2016

Árvores balanceadas

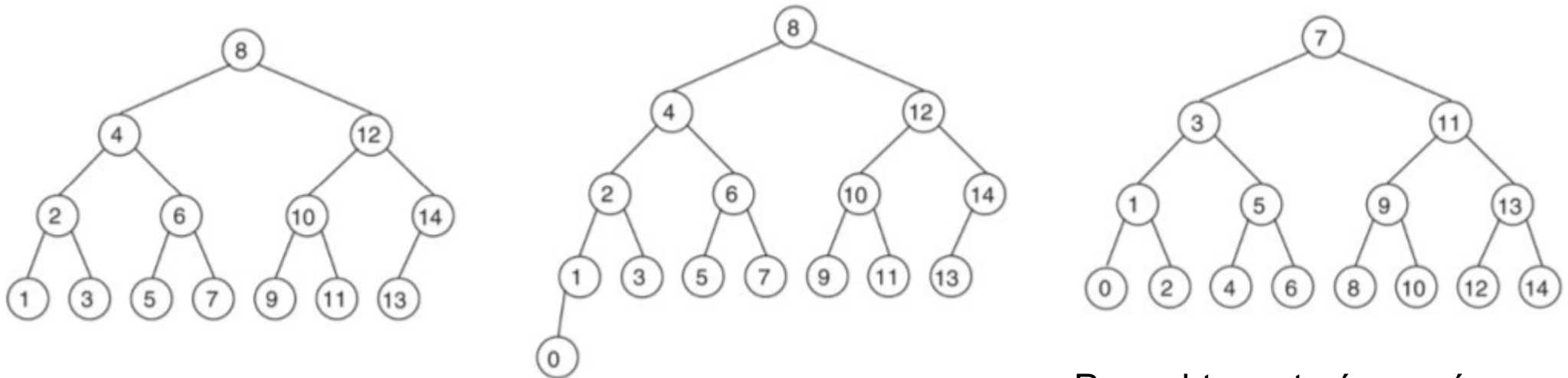
Árvores balanceadas

- As ABB permitem buscar de forma eficiente um elemento dada uma chave.
- Deseja-se que o custo de acesso tenha a ordem de grandeza de uma árvore ótima $\rightarrow O(\lg(n))$
 - Este custo deve-se manter ao longo da utilização da estrutura (inclusive após inserções/remoções).
 - O custo de ter a estrutura balanceada deve estar na, idealmente, mesma ordem de grandeza.

Uma árvore binária é balanceada (ou equilibrada) se, em cada um de seus nós, as subárvores esquerda e direita tiverem aproximadamente a mesma altura.

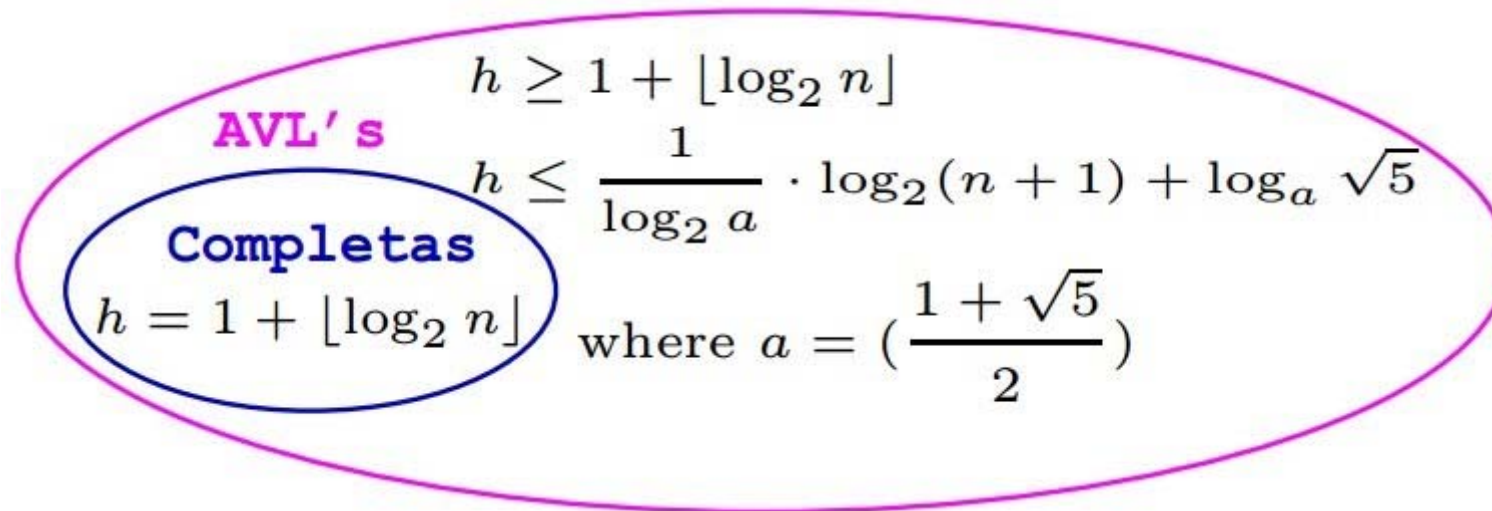
Árvores balanceadas

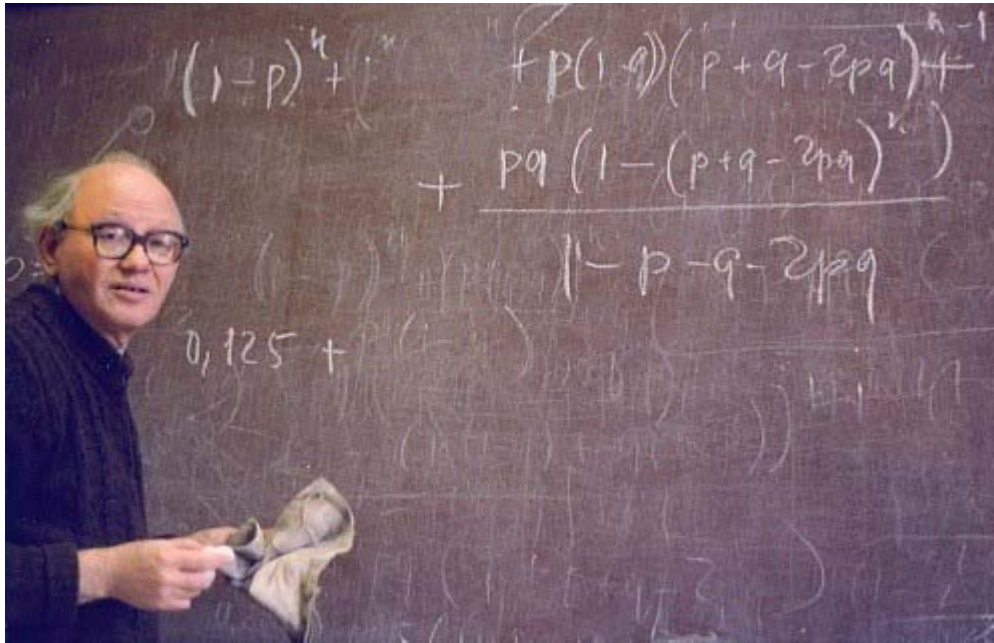
Exemplo ruim para o restabelecimento de árvores completas



Para obter esta árvore é necessário percorrer toda a árvore, $O(n)$

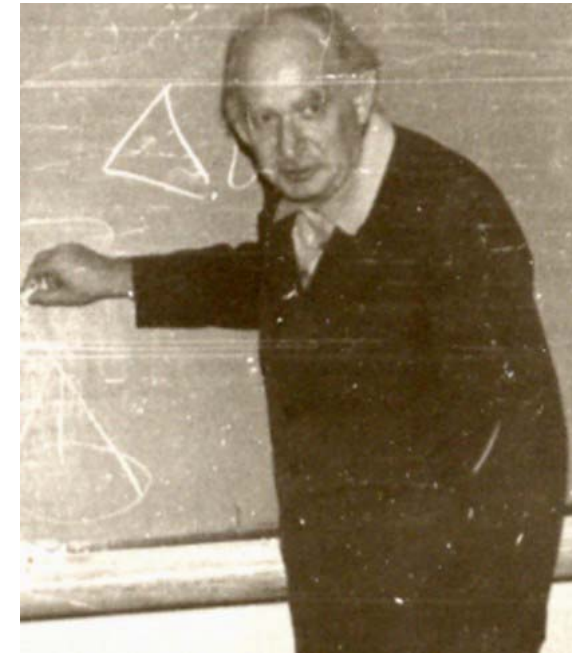
Árvores balanceadas: completas e AVLs





Georgy M. **Adelson-Velsky**
Russia

(1922-2014/abril/26)



Evgenii Mikhailovich **Landis**
Ucrania

(1921-1997)

G.M. Adelson-Velskii y E.M. Landis

“An algorithm for the organization of information”.

Proceedings of the USSR Academy of Sciences, vol. 146, pp. 263–266, **1962**

AN ALGORITHM FOR THE ORGANIZATION OF INFORMATION

G. M. ADEL'SON-VEL'SKIĬ AND E. M. LANDIS

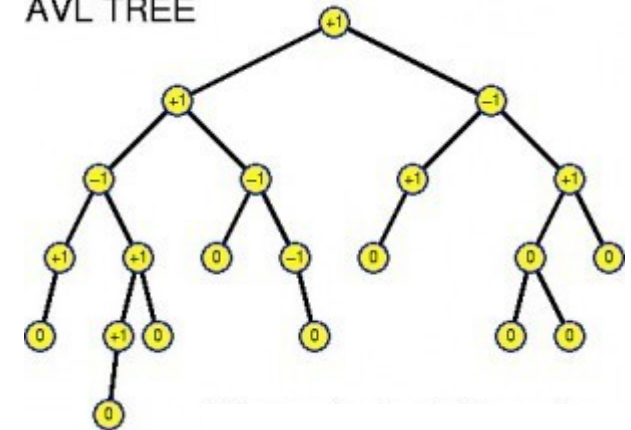
In the present article we discuss the organization of information contained in the cells of an automatic calculating machine. A three-address machine will be used for this study.

Statement of the problem. The information enters a machine in sequence from a certain reserve. The information element is contained in a group of cells which are arranged one after the other. A certain number (the information estimate), which is different for different elements, is contained in the information element. The information must be organized in the memory of the machine in such a way that at any moment a very large number of operations is not required to scan the information with the given evaluation and to record the new information element.

An algorithm is proposed in which both the search and the recording are carried out in $C \lg N$ operations, where N is the number of information elements which have entered at a given moment.

A part of the memory of the machine is set aside to store the incoming information. The information elements are arranged there in their order of entry. Moreover, in another part of the memory a "reference board" [1] is formed, each cell of which corresponds to one of the information elements.

AVL TREE



AVL foi a primeira estrutura (conhecida) de árvore de altura balanceada/equilibrada.

Árvores AVL

- Devido ao balanceamento da altura da árvore, as operações de:

- Busca
- Inserção
- Remoção

em uma árvore com n elementos podem ser efetuadas em $O(\lg n)$ mesmo no pior caso.

Árvores AVL

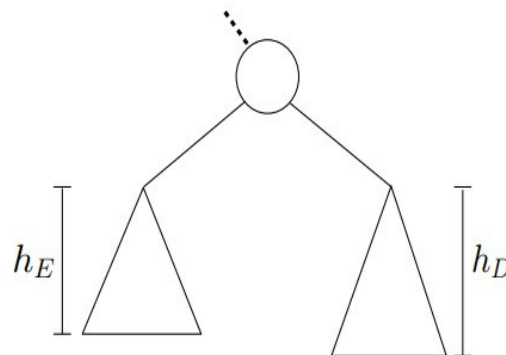
- Devido ao balanceamento da altura da árvore, as operações de:
 - Busca
 - Inserção
 - Remoçãoem uma árvore com n elementos podem ser efetuadas em $O(\lg n)$ mesmo no pior caso.
- Um teorema provado por Adelson-Velskii e Landis **garante** que a árvore balanceada nunca será 45% mais alta que a correspondente árvore perfeitamente balanceada, independentemente do número de nós existentes.

Árvores AVL

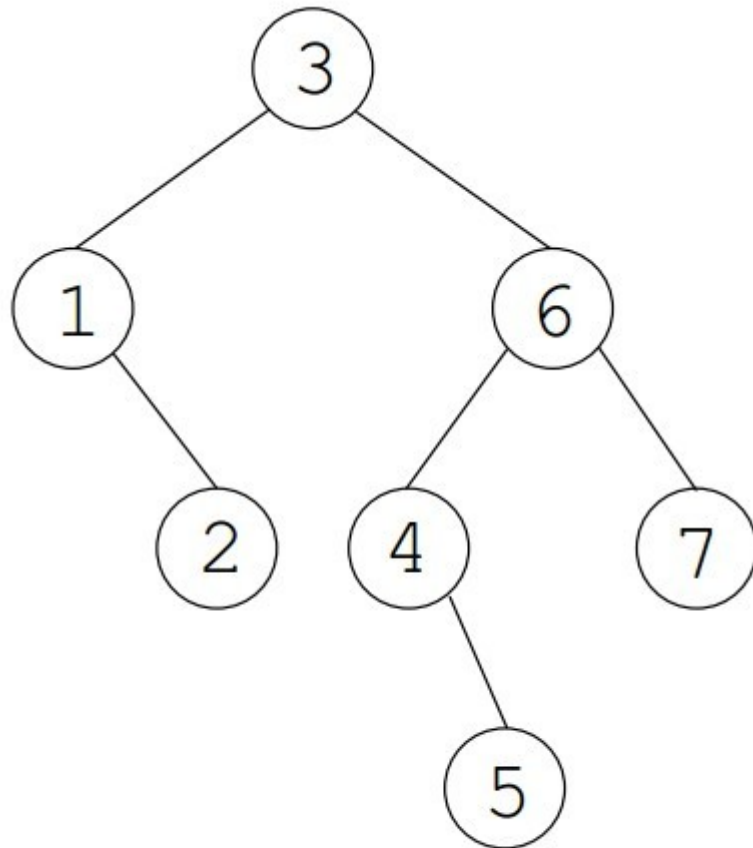
- Uma árvore AVL é definida como:
 - Uma árvore vazia é uma árvore AVL.

Árvores AVL

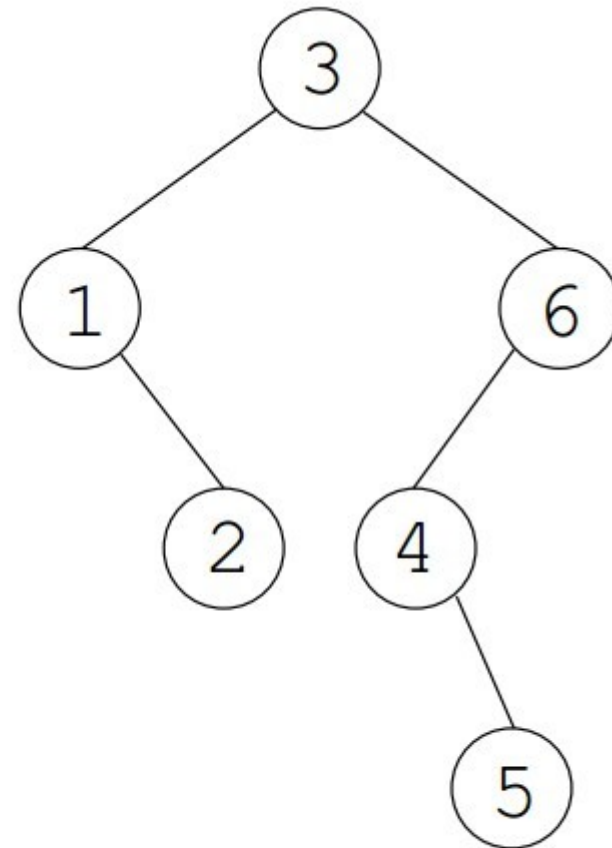
- Uma árvore AVL é definida como:
 - Uma árvore vazia é uma árvore AVL.
 - Sendo T uma ABB, com subárvores esquerda (L) e direita (R), T será uma árvore AVL contanto que:
 - L e R são árvores AVL
 - $|h_L - h_R| \leq 1$
- A definição de uma ABB de altura equilibrada (AVL) requer que cada subárvore seja também de altura equilibrada.



Árvores AVL



AVL



NÃO AVL

Fator de balanceamento

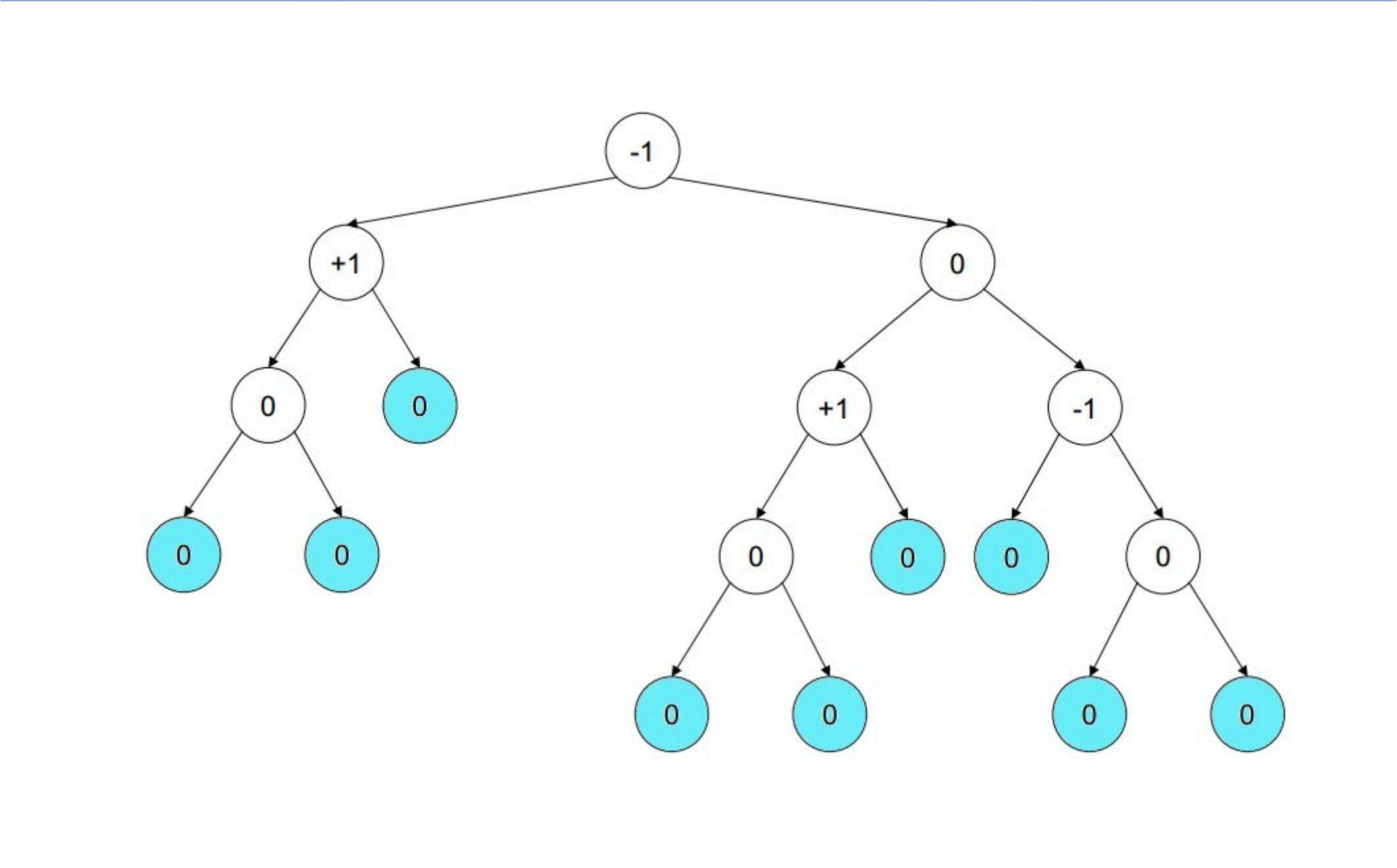
Fator de balanceamento

- O **fator de balanceamento/equilíbrio** de um nó **T** em uma ABB é definido como: $h_L - h_R$

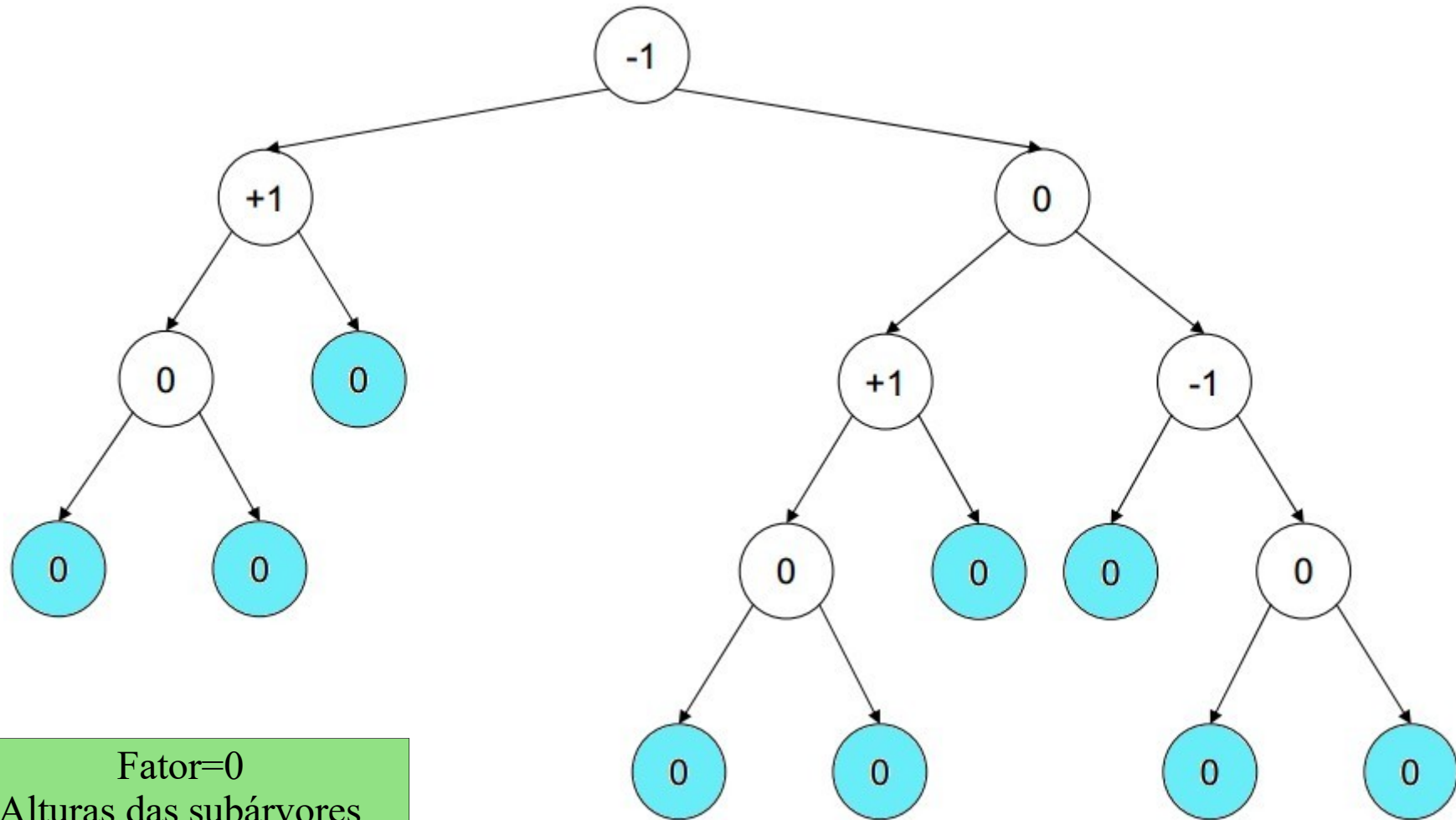
Fator de balanceamento

- O **fator de balanceamento/equilíbrio** de um nó **T** em uma ABB é definido como: $h_L - h_R$
- Para qualquer nó **T** em uma árvore AVL, o fator de balanceamento assume o valor: **+1, 0, -1**.
- O fator de balanceamento de uma folha?

Fator de balanceamento

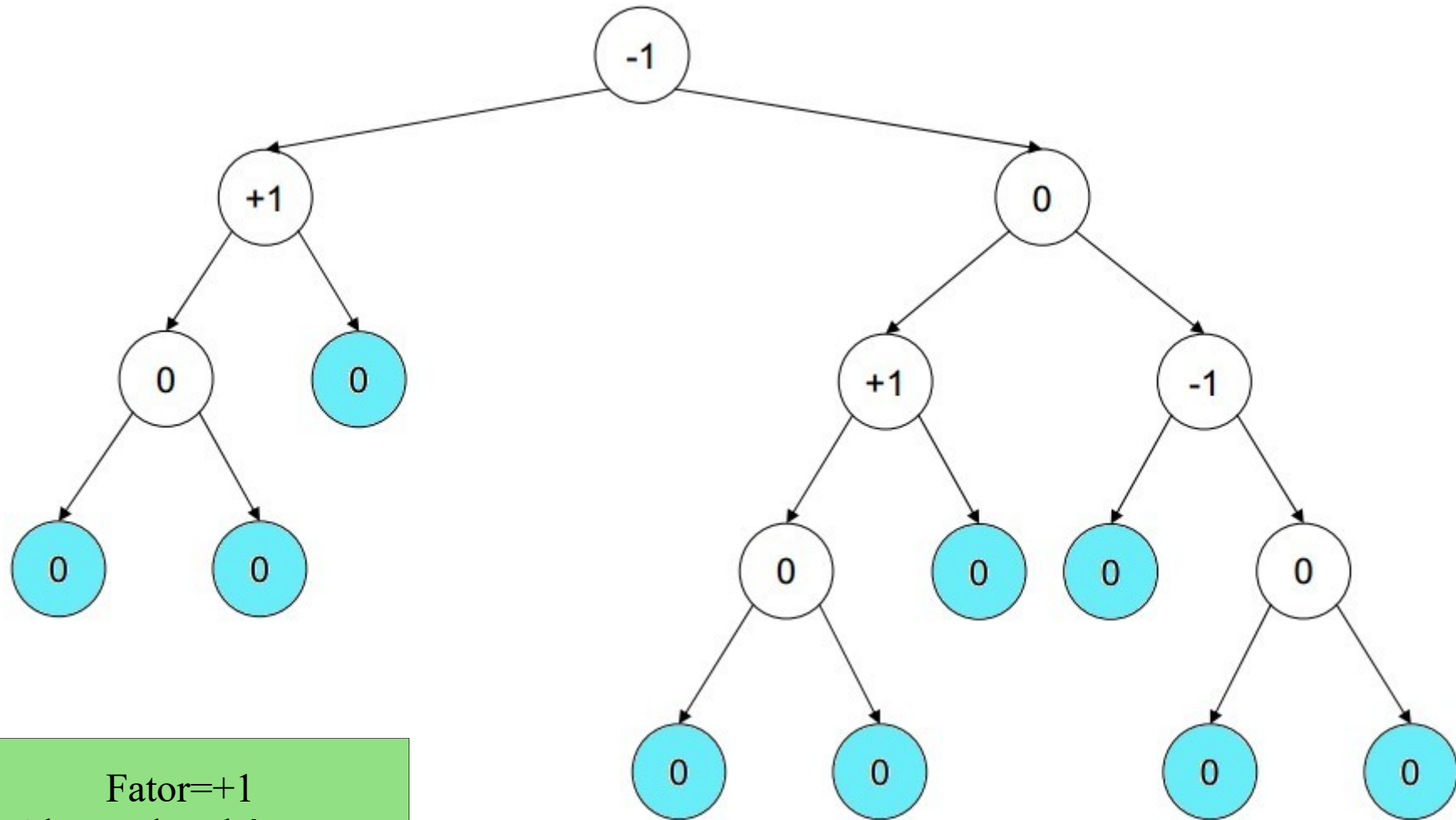


Fator de balanceamento



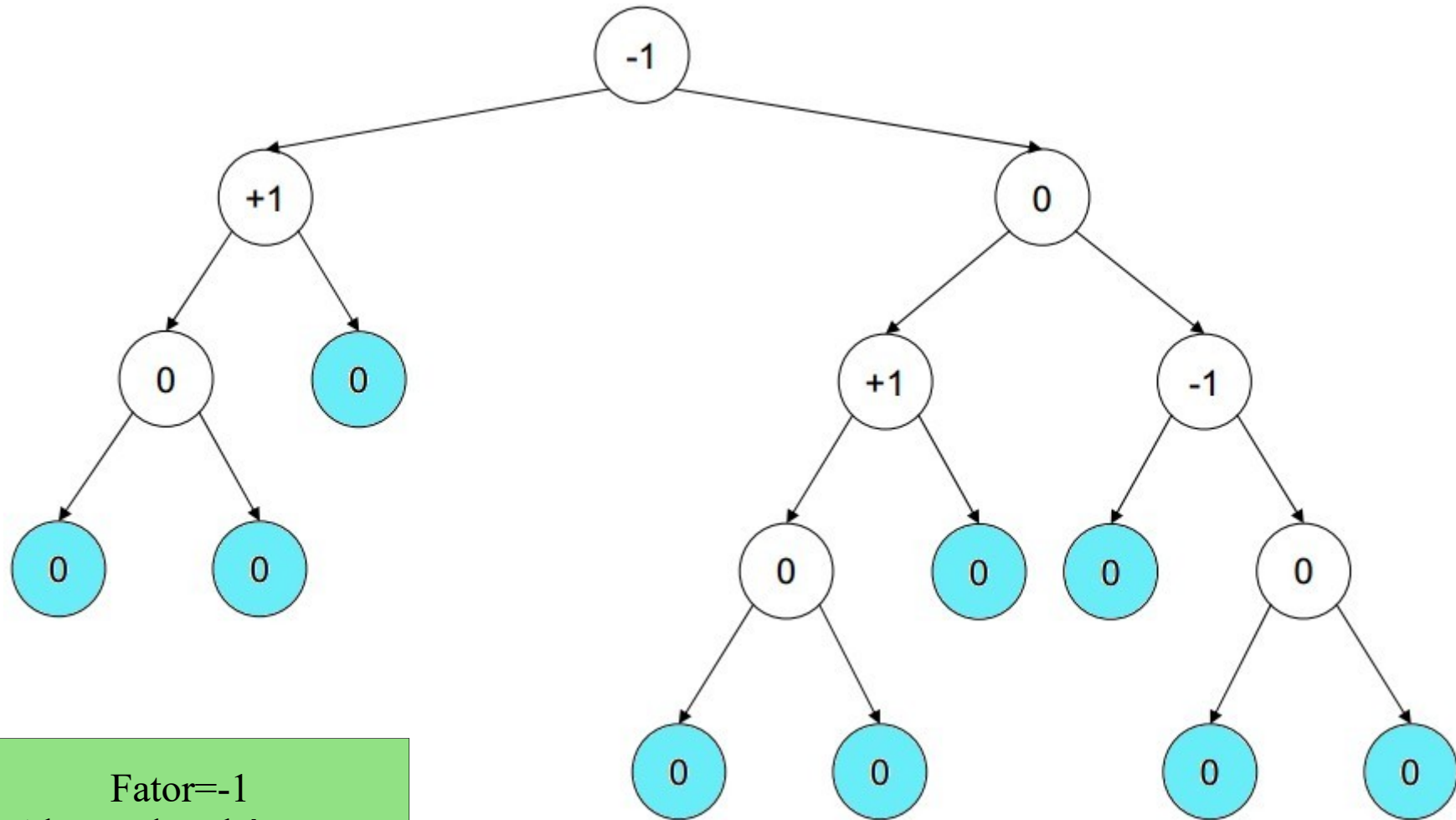
Fator=0
Alturas das subárvores
esquerda e direita
são iguais

Fator de balanceamento



Fator=+1
Alturas da subárvore
esquerda é maior

Fator de balanceamento



Fator=-1
Alturas da subárvore
esquerda é menor

Exemplo: Inserção de 'Maio'

Depois da inserção

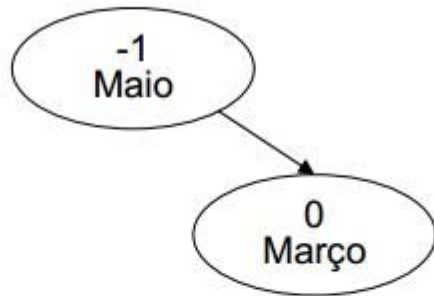


Depois do rebalanceamento

*Sem necessidade
de rebalanceamento*

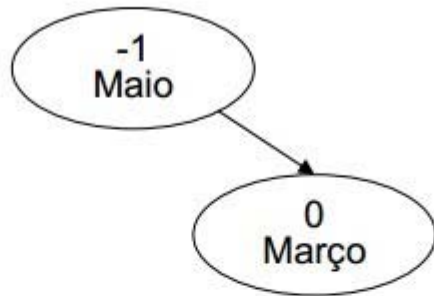
Exemplo: Inserção de 'Março'

Depois da inserção



Exemplo: Inserção de 'Março'

Depois da inserção

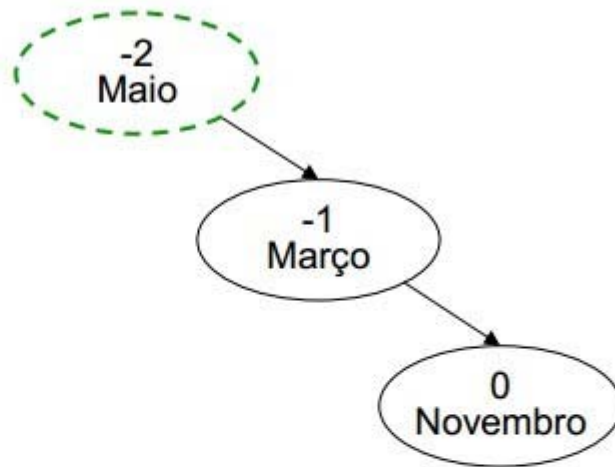


Depois do rebalanceamento

*Sem necessidade
de rebalanceamento*

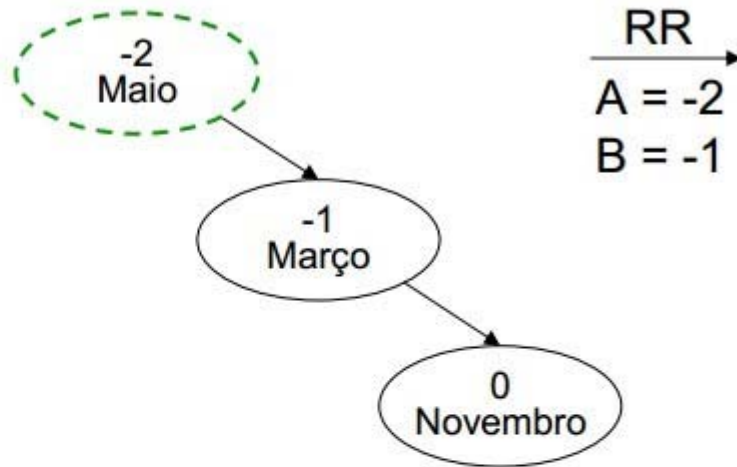
Exemplo: Inserção de 'Novembro'

Depois da inserção



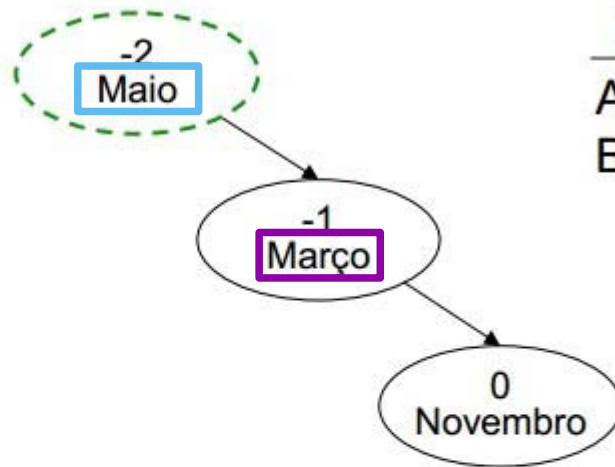
Exemplo: Inserção de 'Novembro'

Depois da inserção



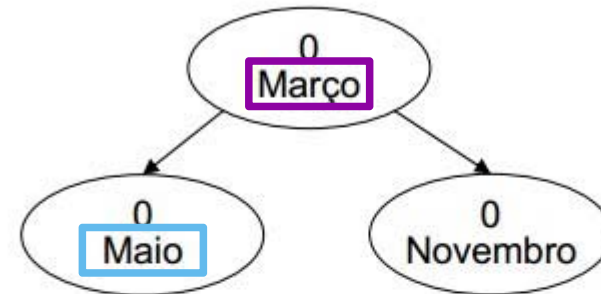
Exemplo: Inserção de 'Novembro'

Depois da inserção



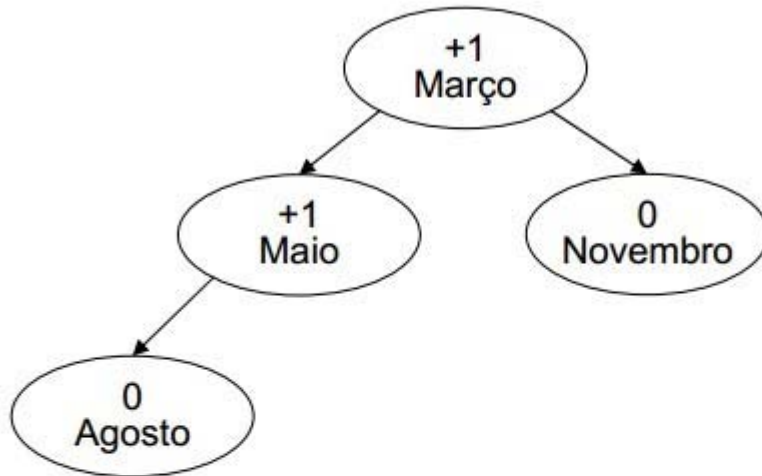
RR
A = -2
B = -1

Depois do rebalanceamento



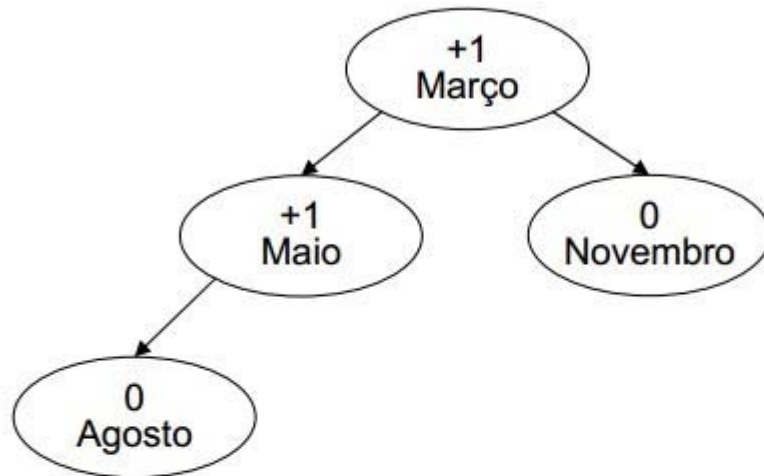
Exemplo: Inserção de 'Agosto'

Depois da inserção



Exemplo: Inserção de 'Agosto'

Depois da inserção

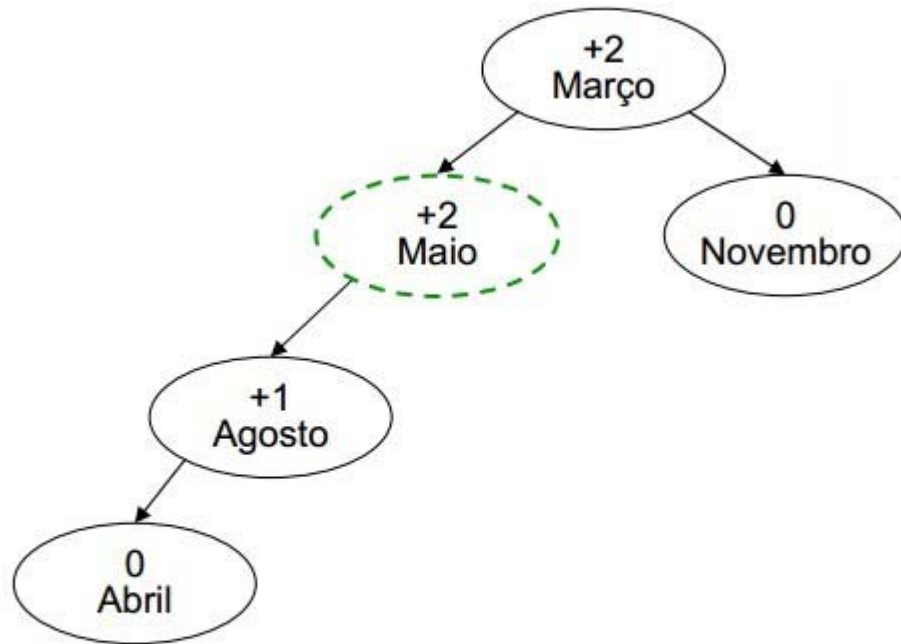


Depois do rebalanceamento

*Sem necessidade
de rebalanceamento*

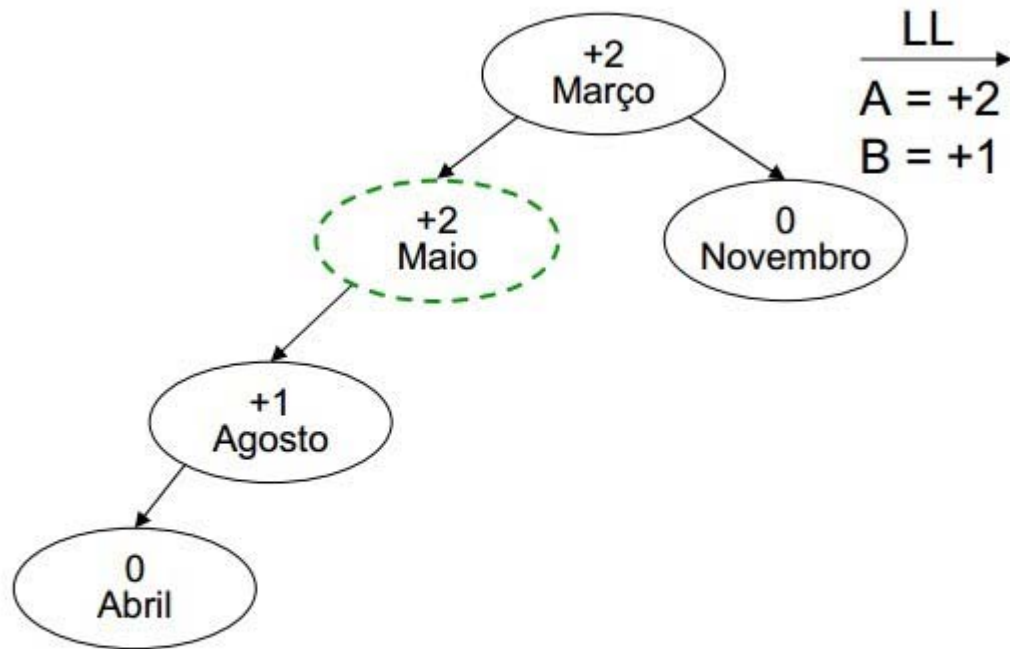
Exemplo: Inserção de 'Abril'

Depois da inserção



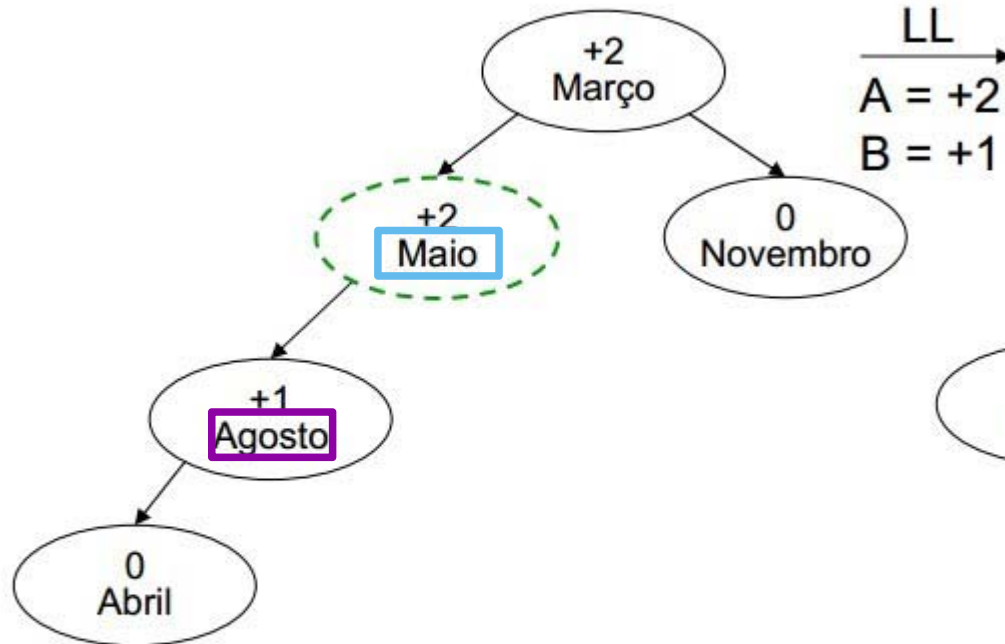
Exemplo: Inserção de 'Abril'

Depois da inserção

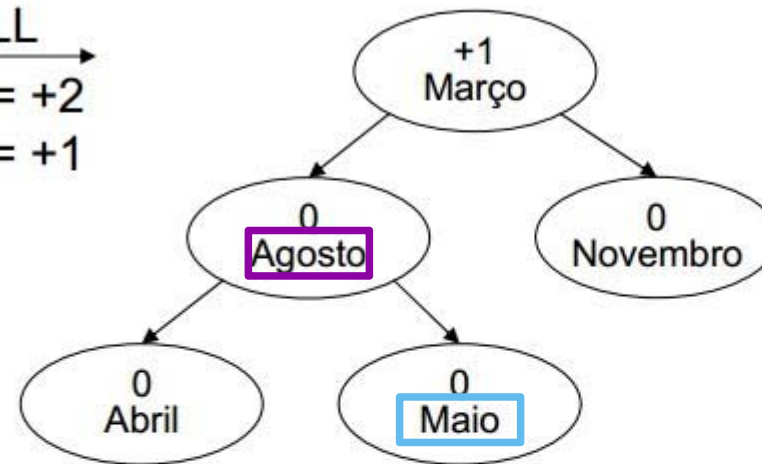


Exemplo: Inserção de 'Abril'

Depois da inserção

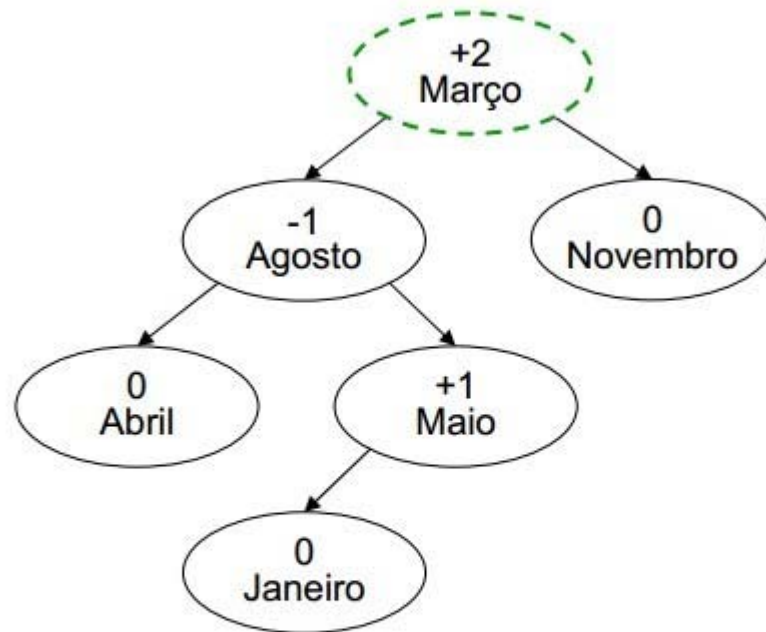


Depois do rebalanceamento



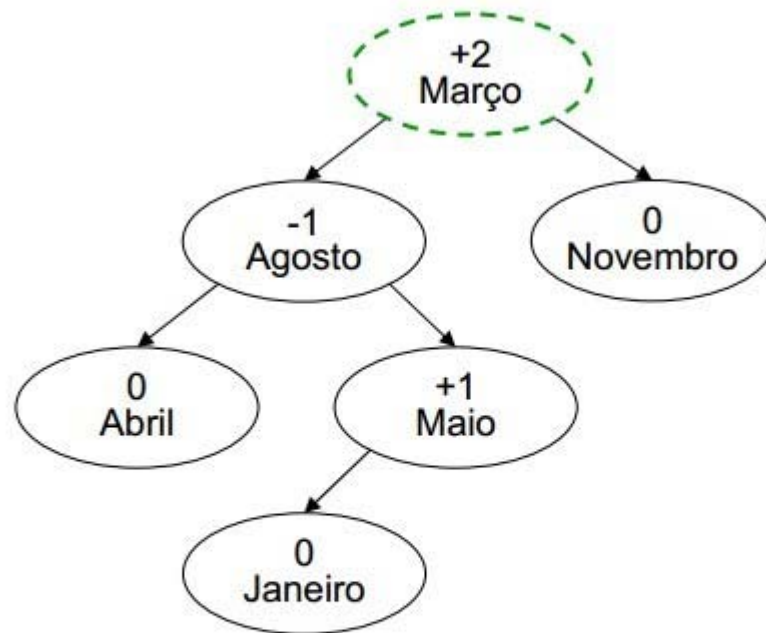
Exemplo: Inserção de 'Janeiro'

Depois da inserção



Exemplo: Inserção de 'Janeiro'

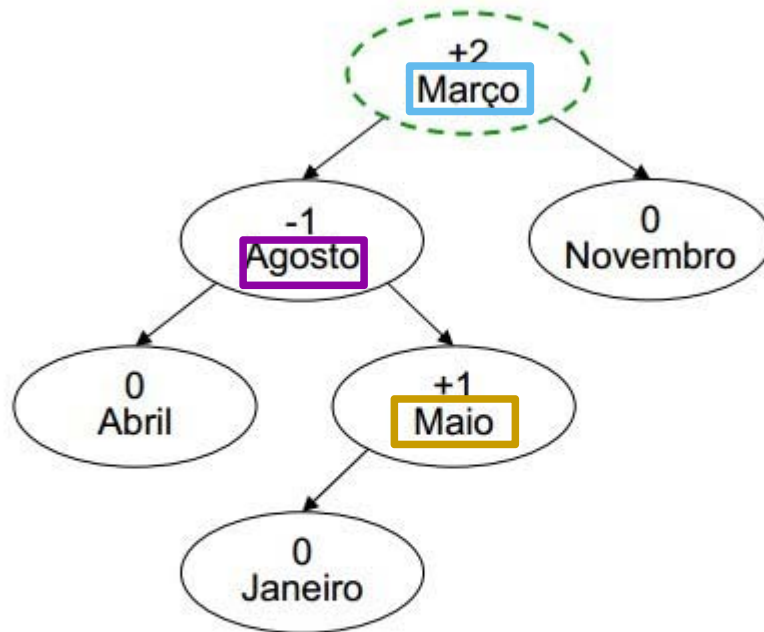
Depois da inserção



LR
A = +2
B = -1

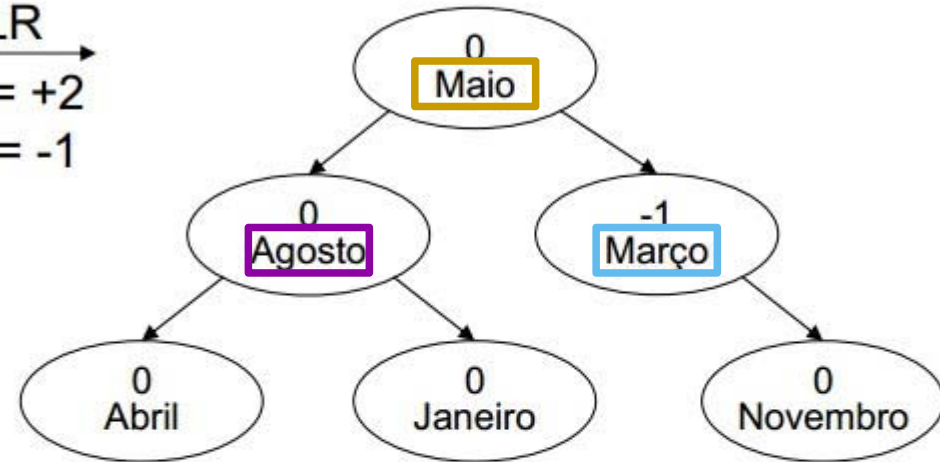
Exemplo: Inserção de 'Janeiro'

Depois da inserção



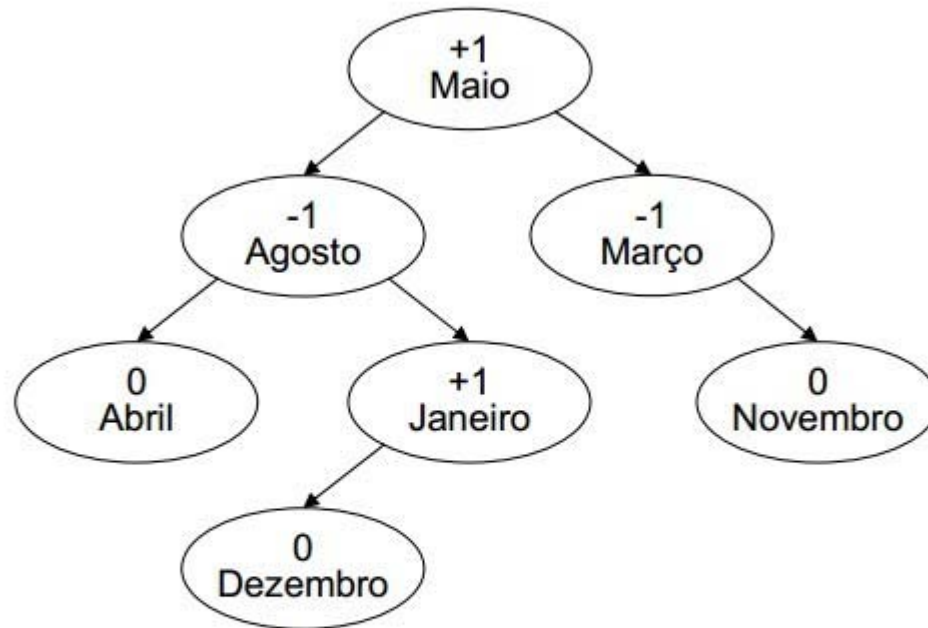
Depois do rebalanceamento

LR
A = +2
B = -1



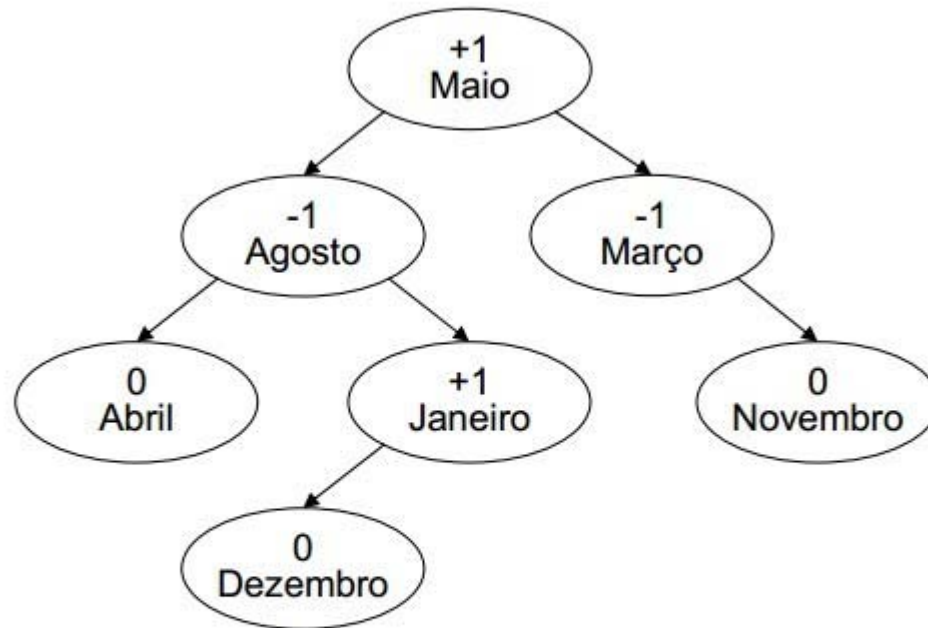
Exemplo: Inserção de 'Dezembro'

Depois da inserção



Exemplo: Inserção de 'Dezembro'

Depois da inserção

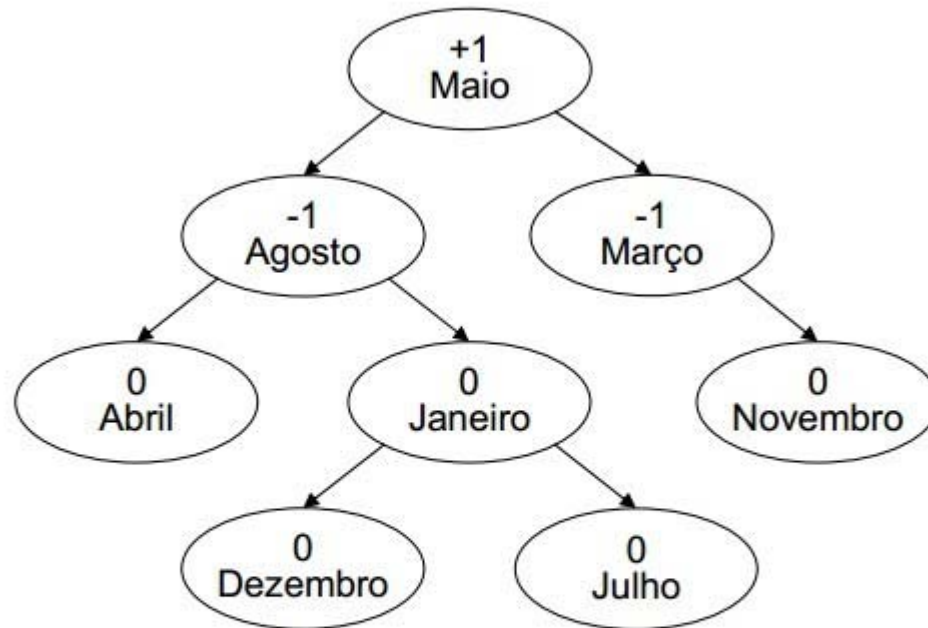


Depois do rebalanceamento

*Sem necessidade
de rebalanceamento*

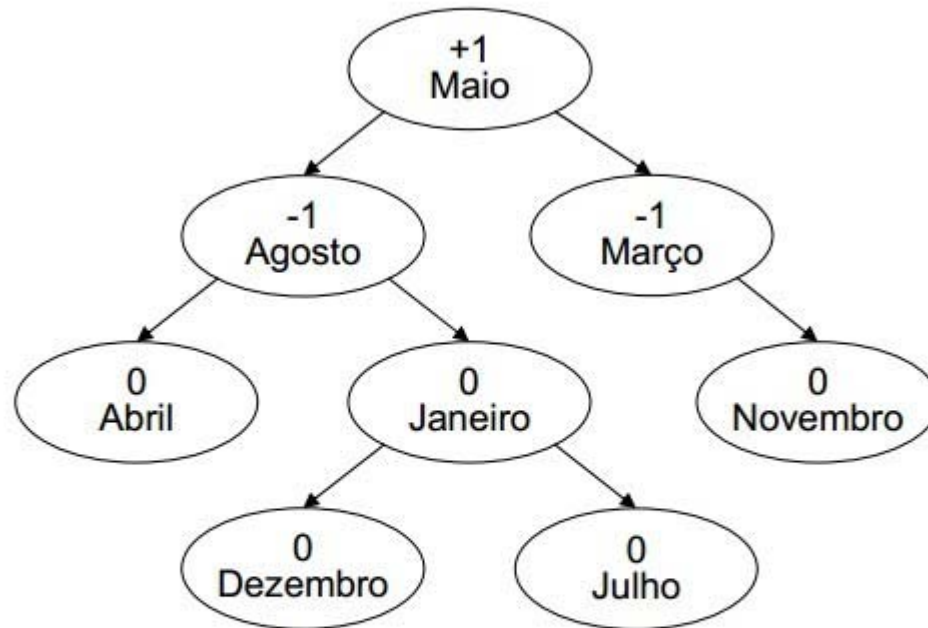
Exemplo: Inserção de 'Julho'

Depois da inserção



Exemplo: Inserção de 'Julho'

Depois da inserção

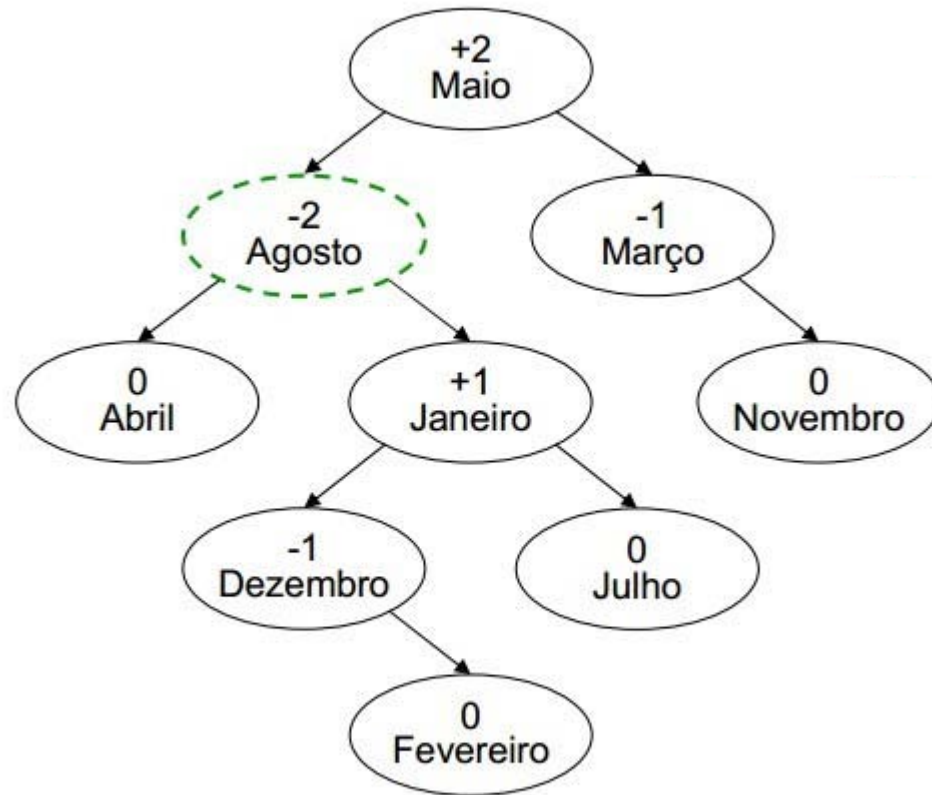


Depois do rebalanceamento

*Sem necessidade
de rebalanceamento*

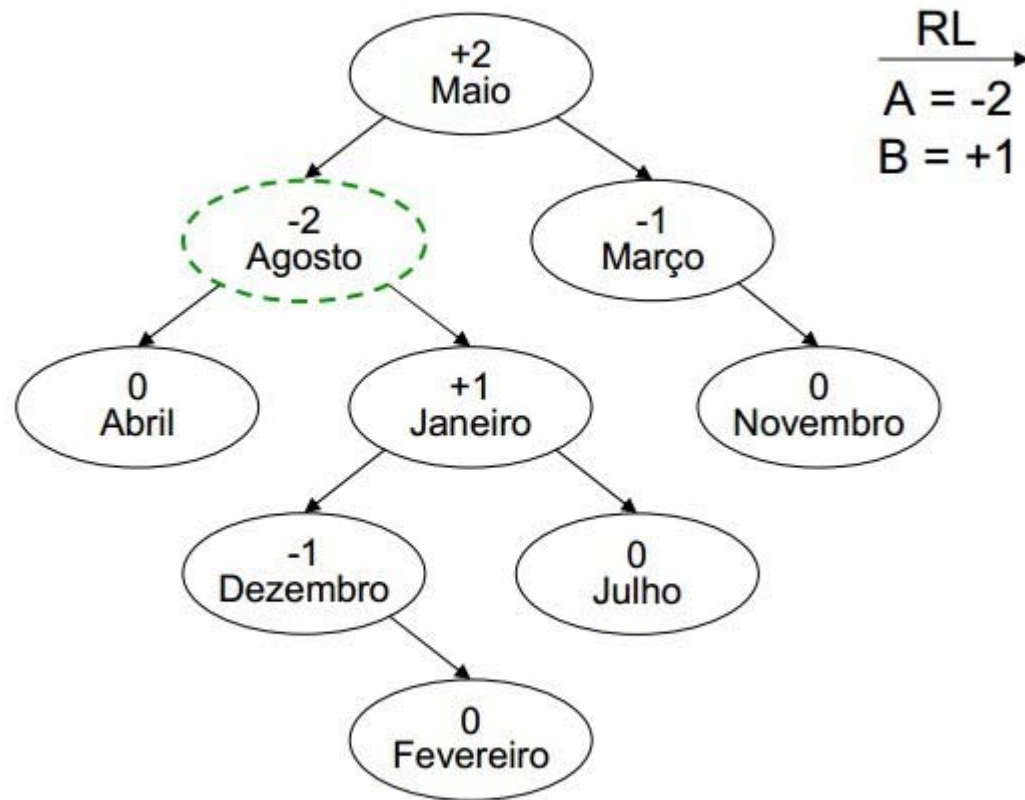
Exemplo: Inserção de 'Fevereiro'

Depois da inserção



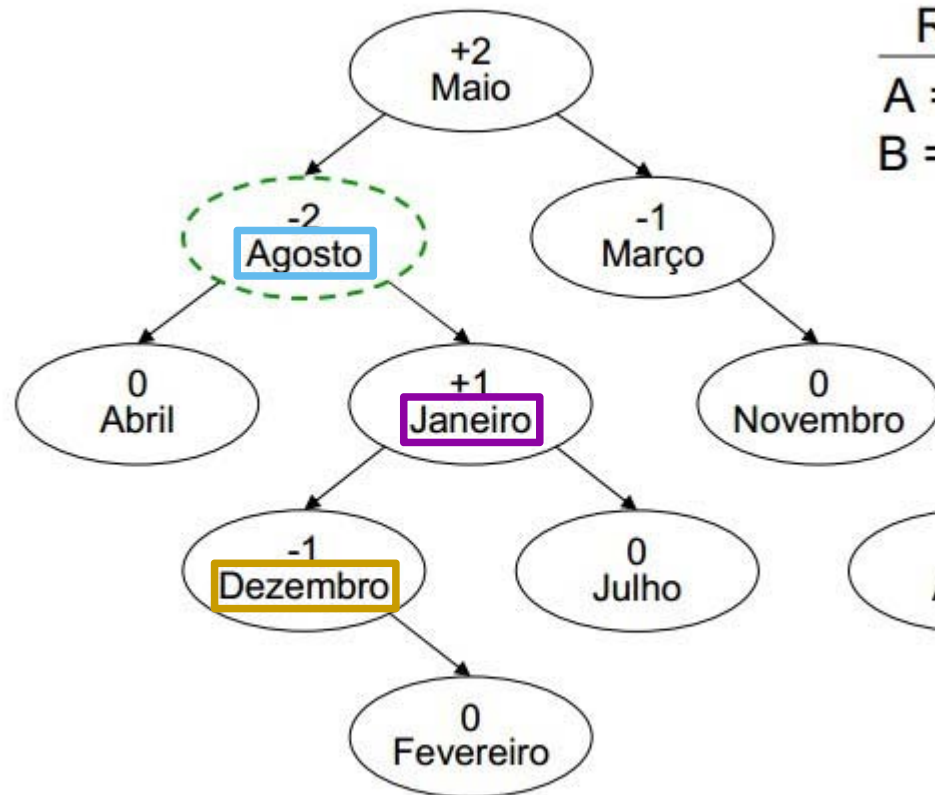
Exemplo: Inserção de 'Fevereiro'

Depois da inserção



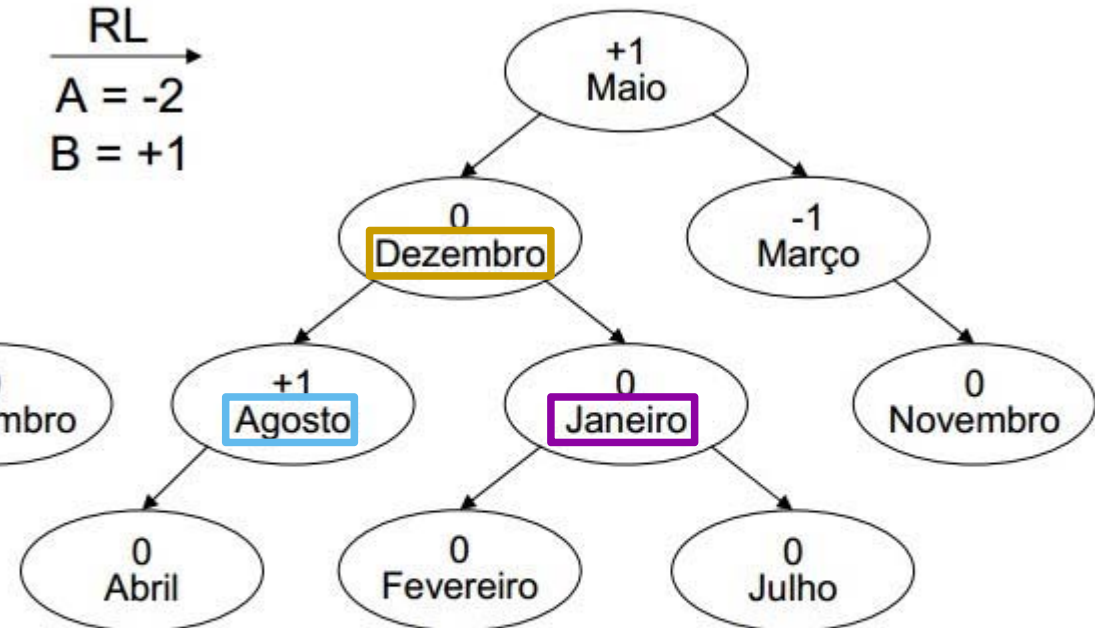
Exemplo: Inserção de 'Fevereiro'

Depois da inserção



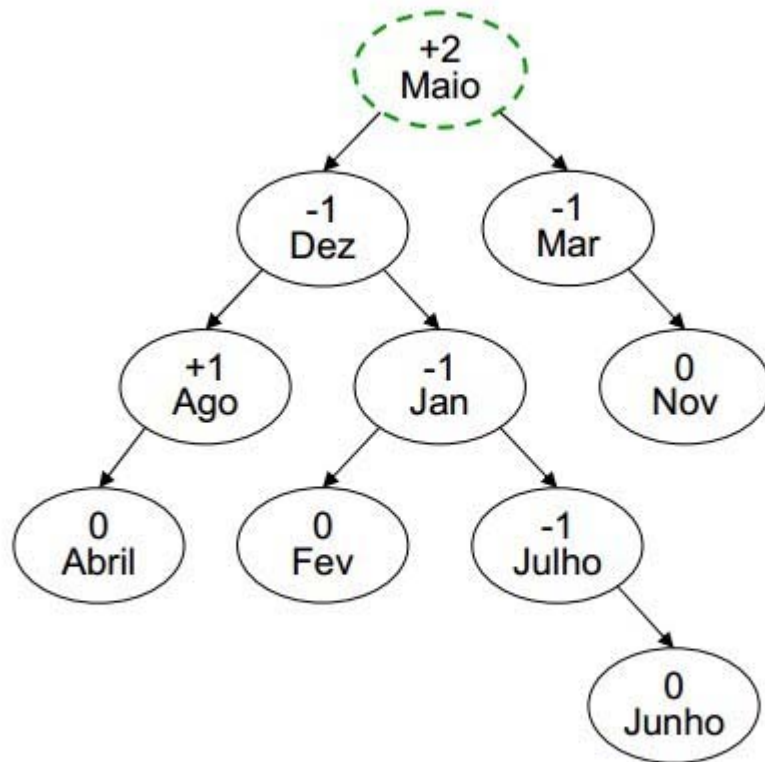
Depois do rebalanceamento

RL
A = -2
B = +1



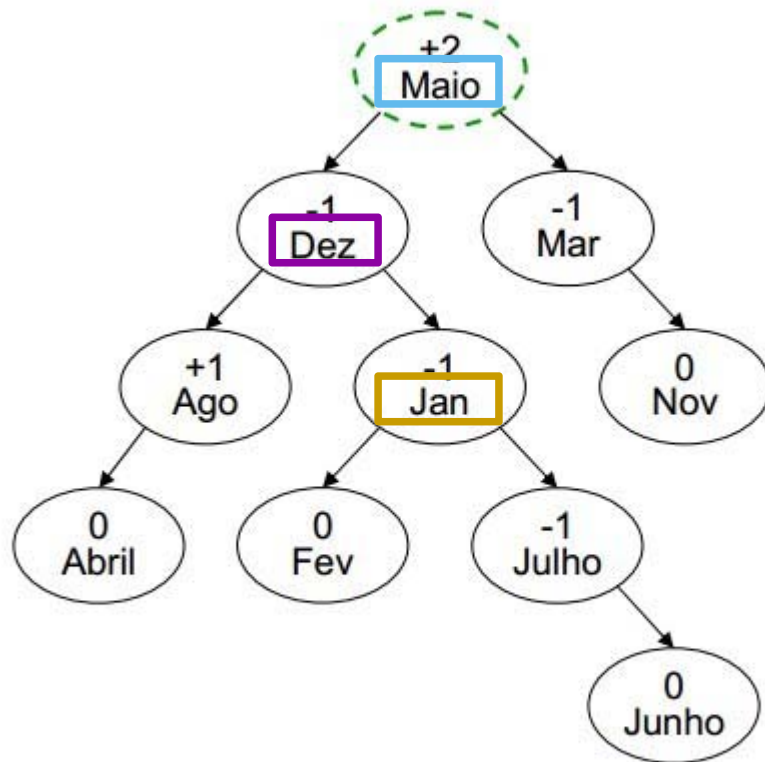
Exemplo: Inserção de 'Junho'

Depois da inserção

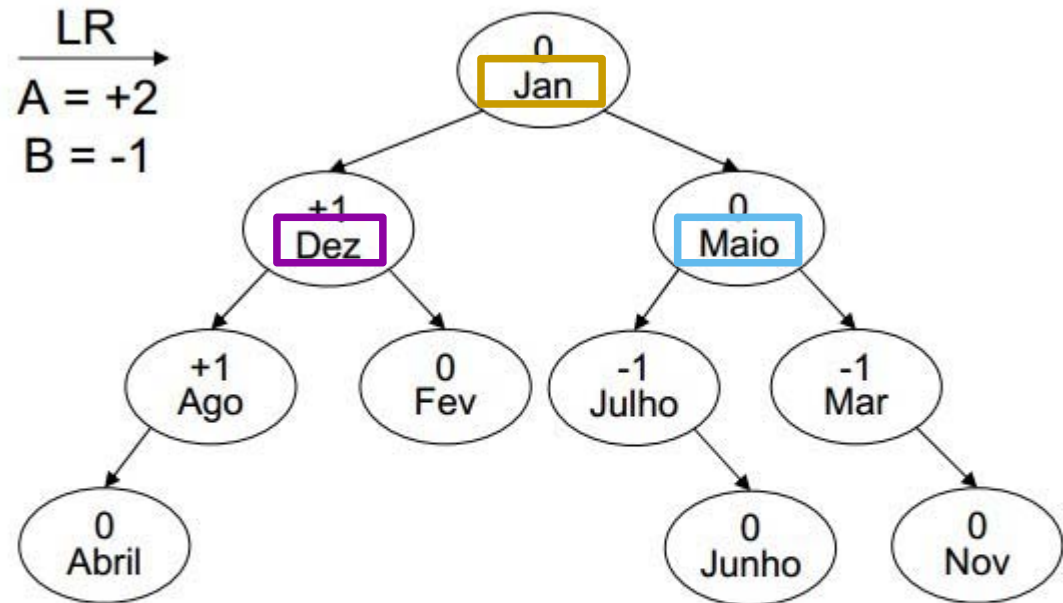


Exemplo: Inserção de 'Junho'

Depois da inserção

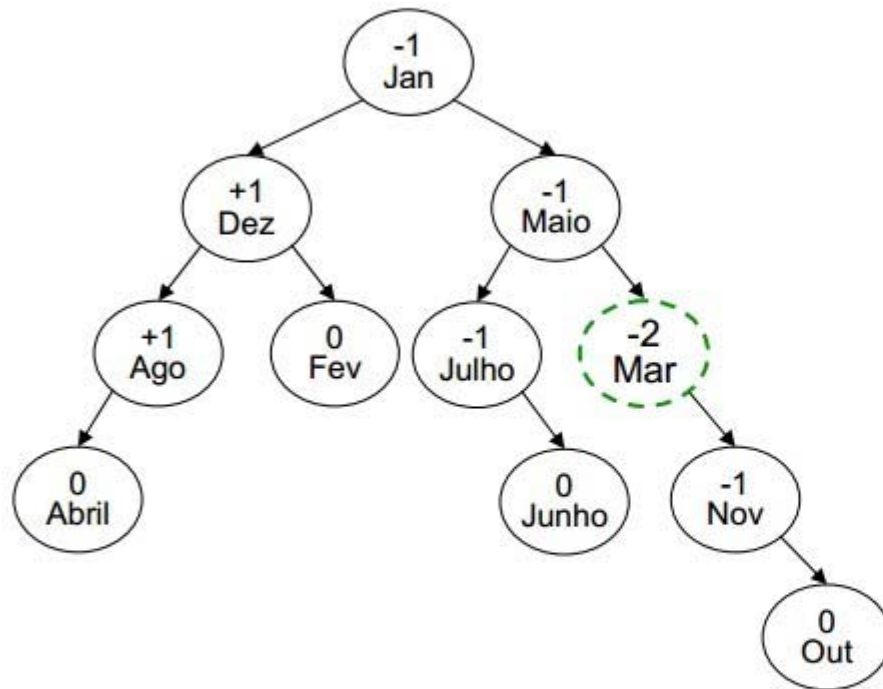


Depois do rebalanceamento



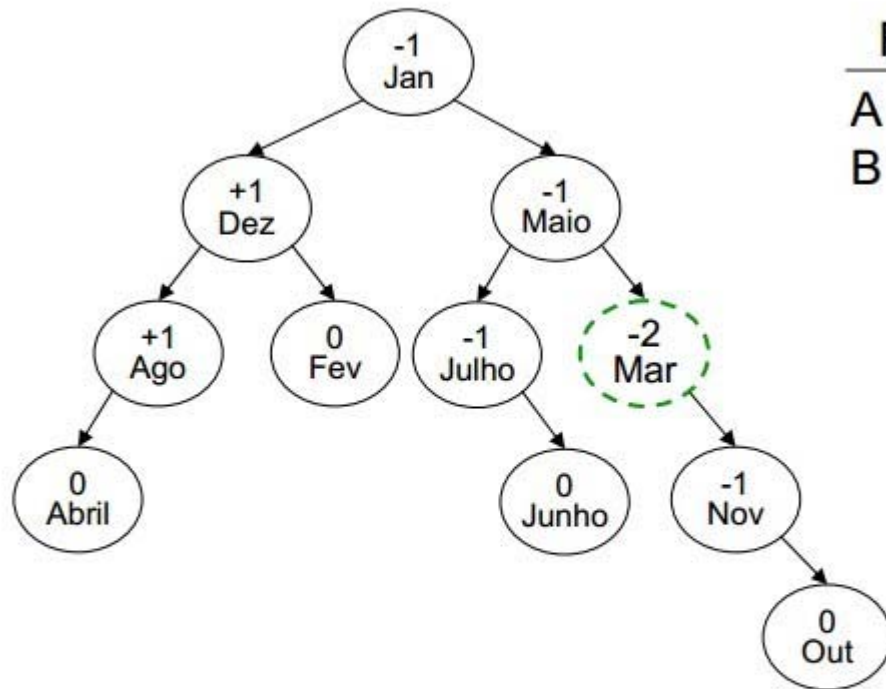
Exemplo: Inserção de 'Outubro'

Depois da inserção



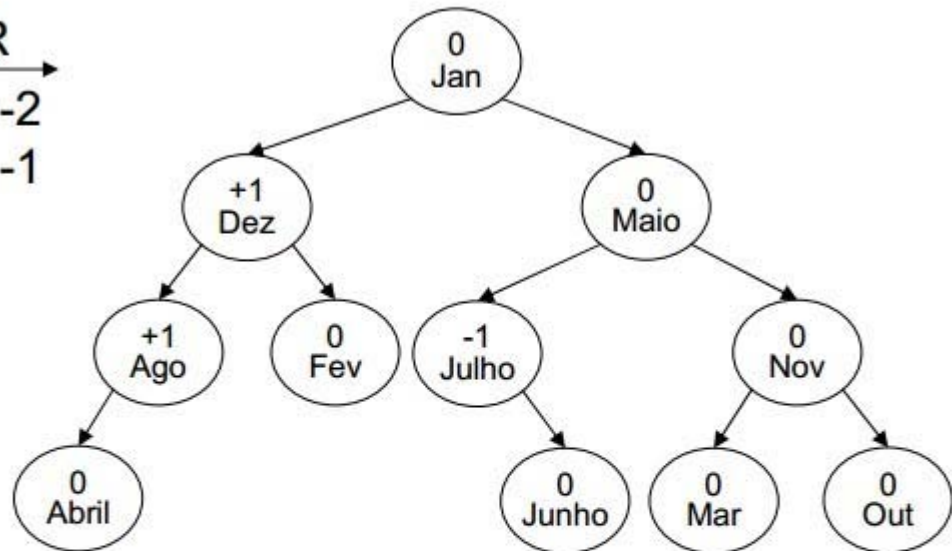
Exemplo: Inserção de 'Outubro'

Depois da inserção



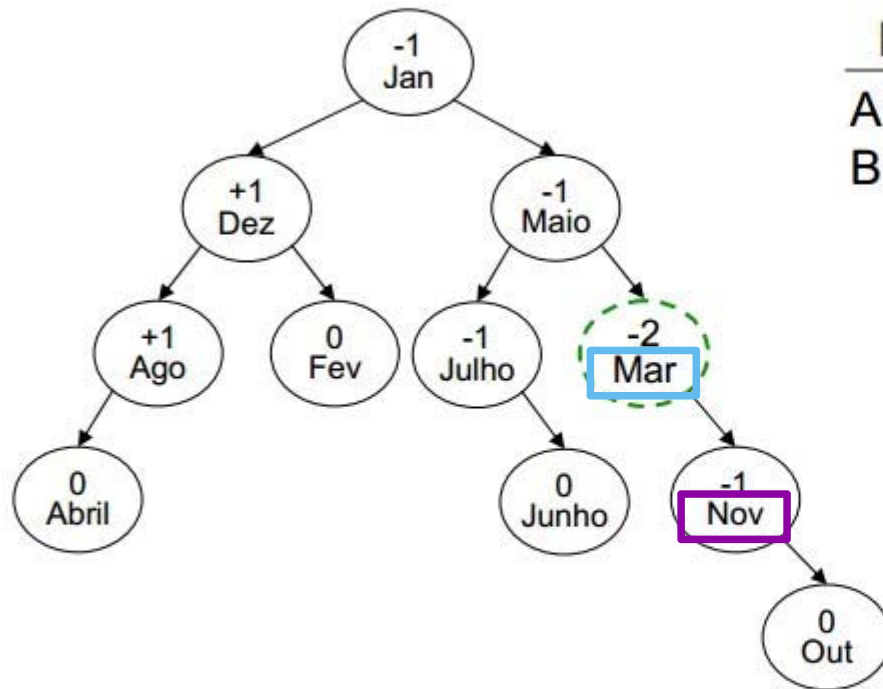
Depois do rebalanceamento

RR
A = -2
B = -1



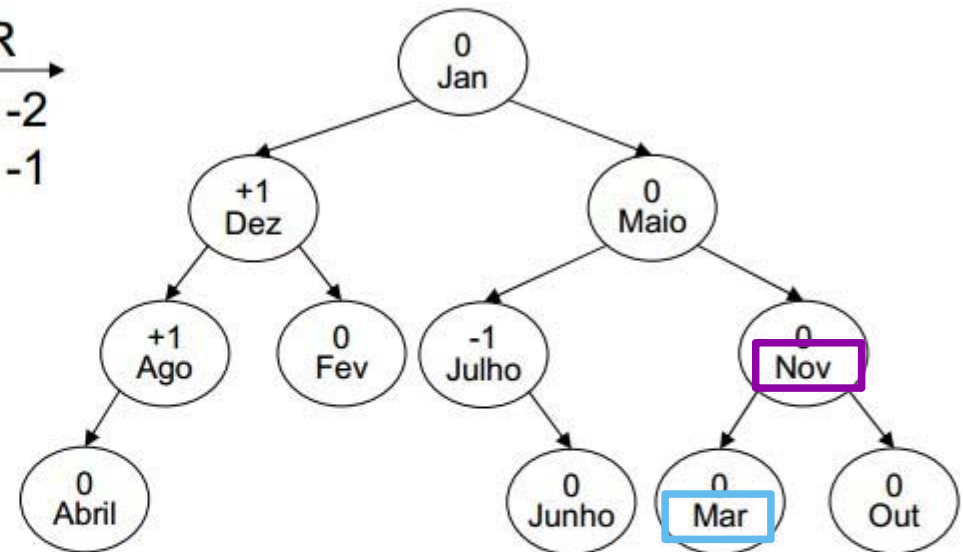
Exemplo: Inserção de 'Outubro'

Depois da inserção



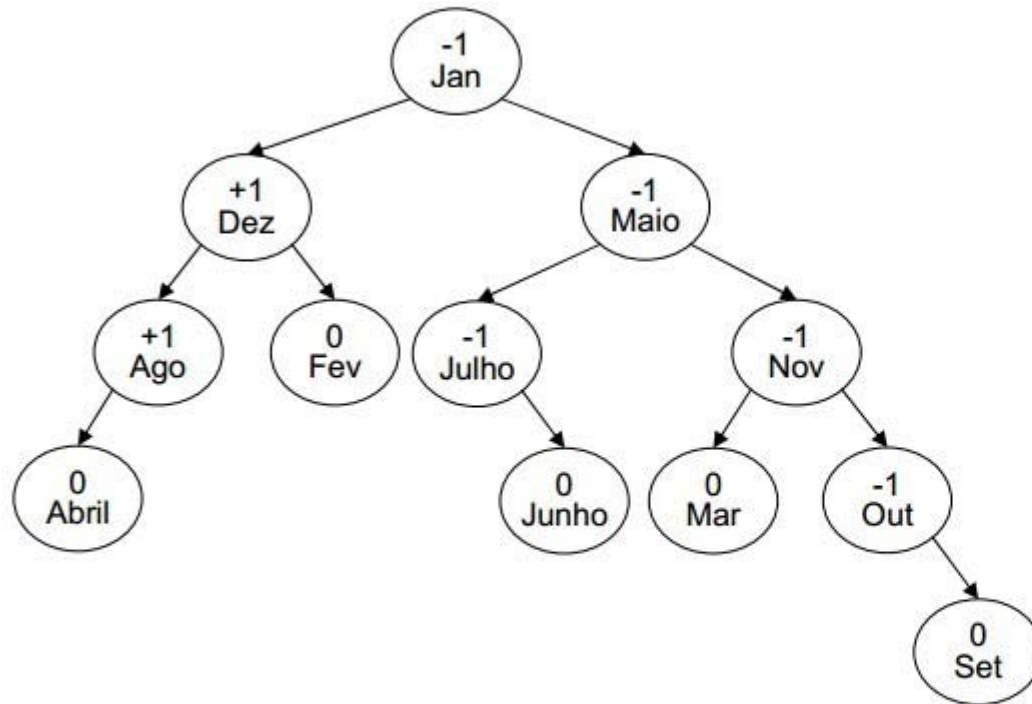
RR
A = -2
B = -1

Depois do rebalanceamento



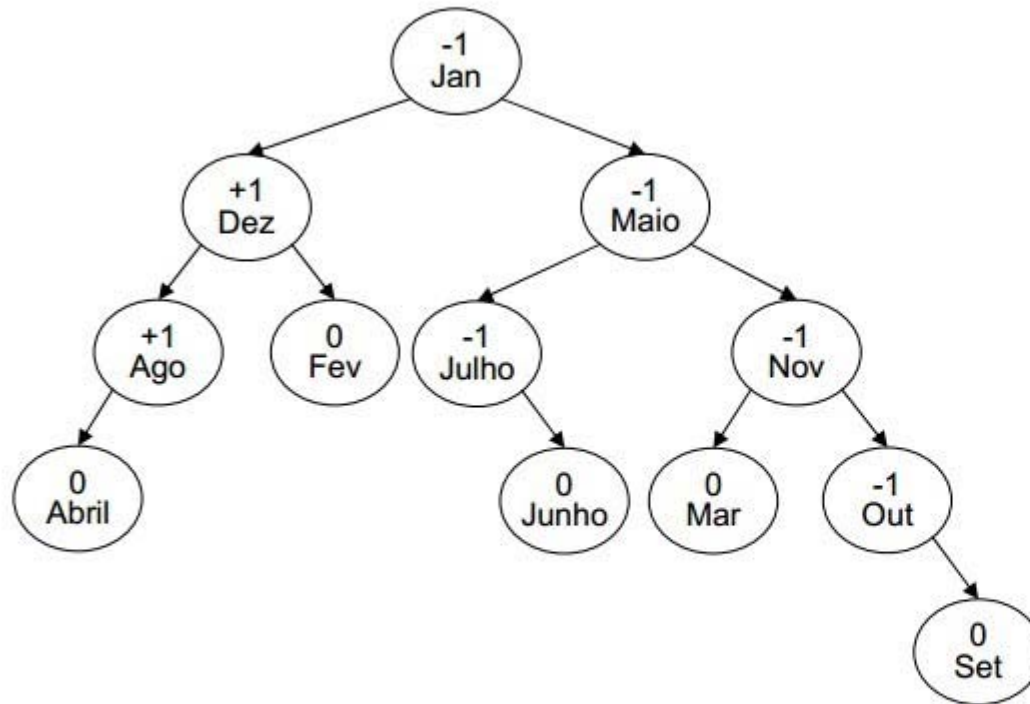
Exemplo: Inserção de 'Setembro'

Depois da inserção



Exemplo: Inserção de 'Setembro'

Depois da inserção



Depois do rebalanceamento

*Sem necessidade
de rebalanceamento*

Rotações

Rotações

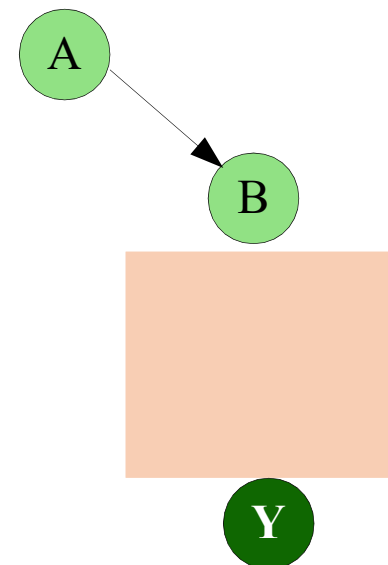
- O processo de rebalanceamento é conduzido utilizando 4 tipos de rotações
 - LL
 - RR
 - LR
 - RL
- Suponha que o novo nó inserido é **Y**:
As rotações são caracterizadas pelo ancestral **A** (com fator de balanceamento **+2** ou **-2**) mais próximo do nó **Y**.

Rotações

- **LL:** Y inserido na subárvore esquerda da subárvore esquerda de A
- **LR:** Y inserido na subárvore direita da subárvore esquerda de A
- **RR:** Y inserido na subárvore direita da subárvore direita de A
- **RL:** Y inserido na subárvore esquerda da subárvore direita de A

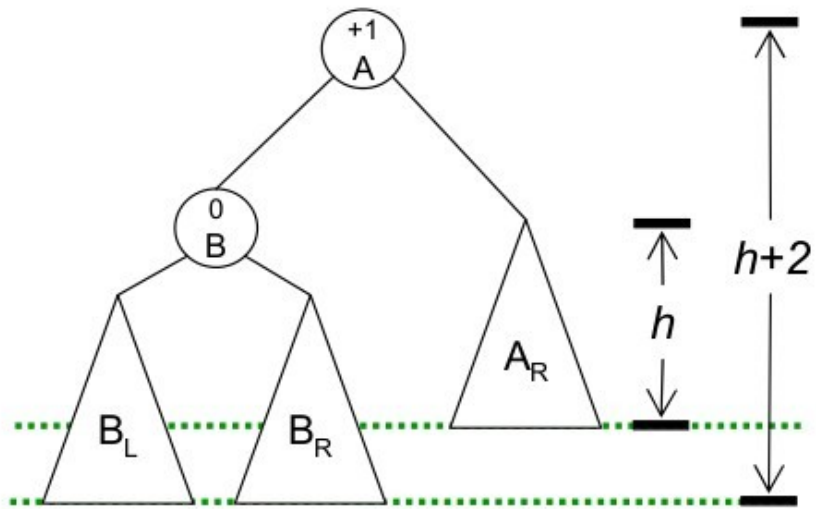
Seja **B** o filho de **A** no qual ocorreu a inserção de **Y**

- **LL** ($A = +2$; $B = +1$)
- **LR** ($A = +2$; $B = -1$)
- **RR** ($A = -2$; $B = -1$)
- **RL** ($A = -2$; $B = +1$)



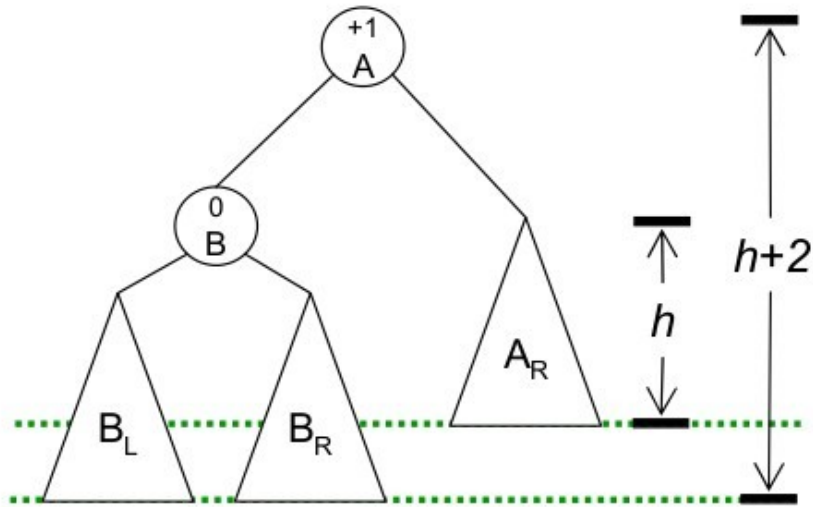
Rotação LL

Subárvore balanceada

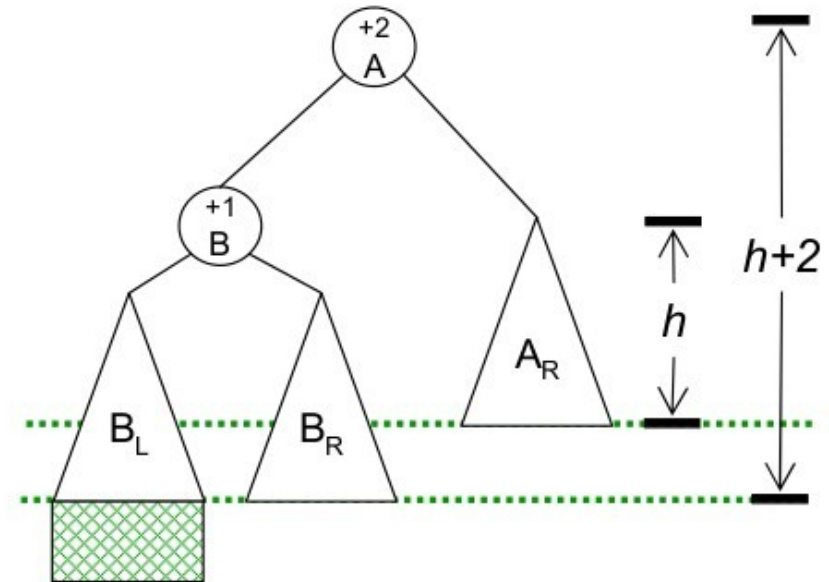


Rotação LL

Subárvore balanceada



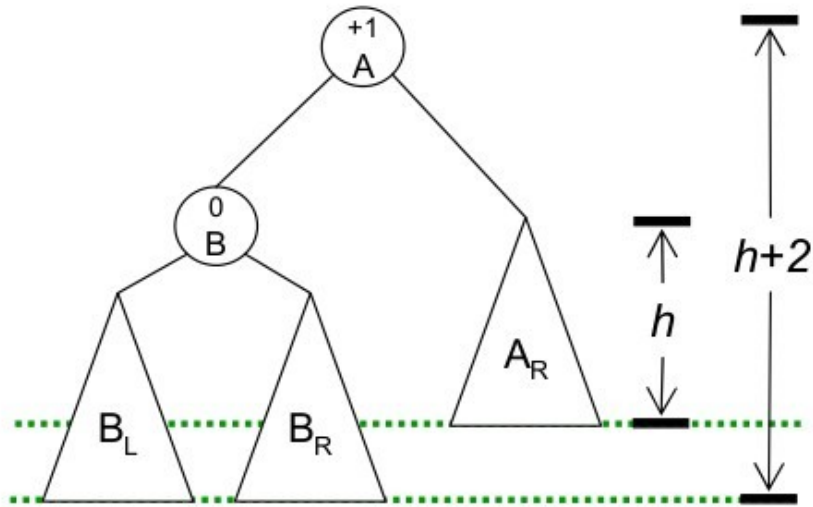
Subárvore desbalanceada após inserção



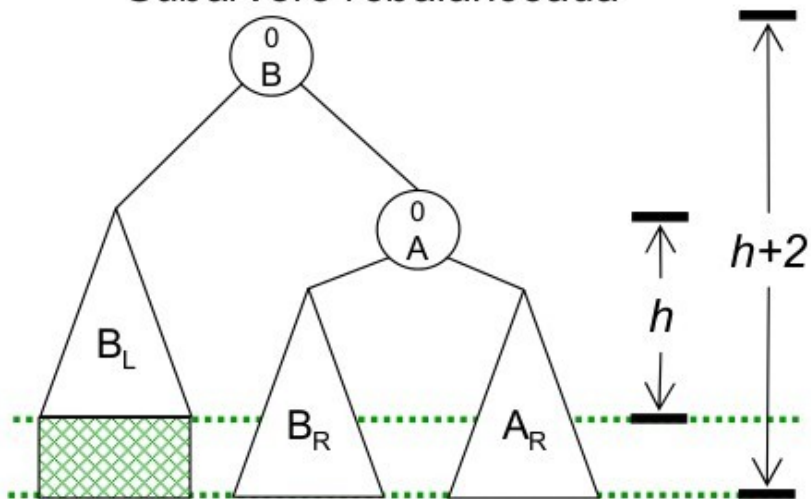
Altura de B_L aumenta para $h+1$

Rotação LL

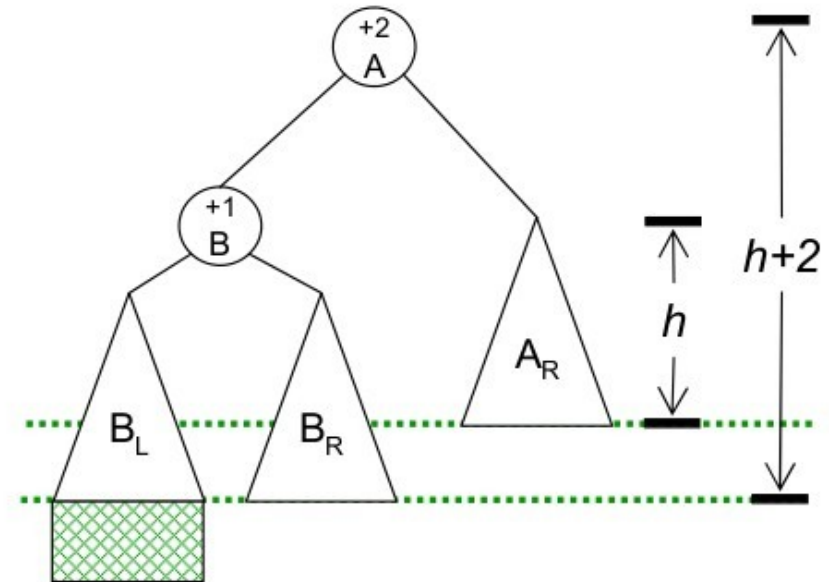
Subárvore balanceada



Subárvore rebalanceada

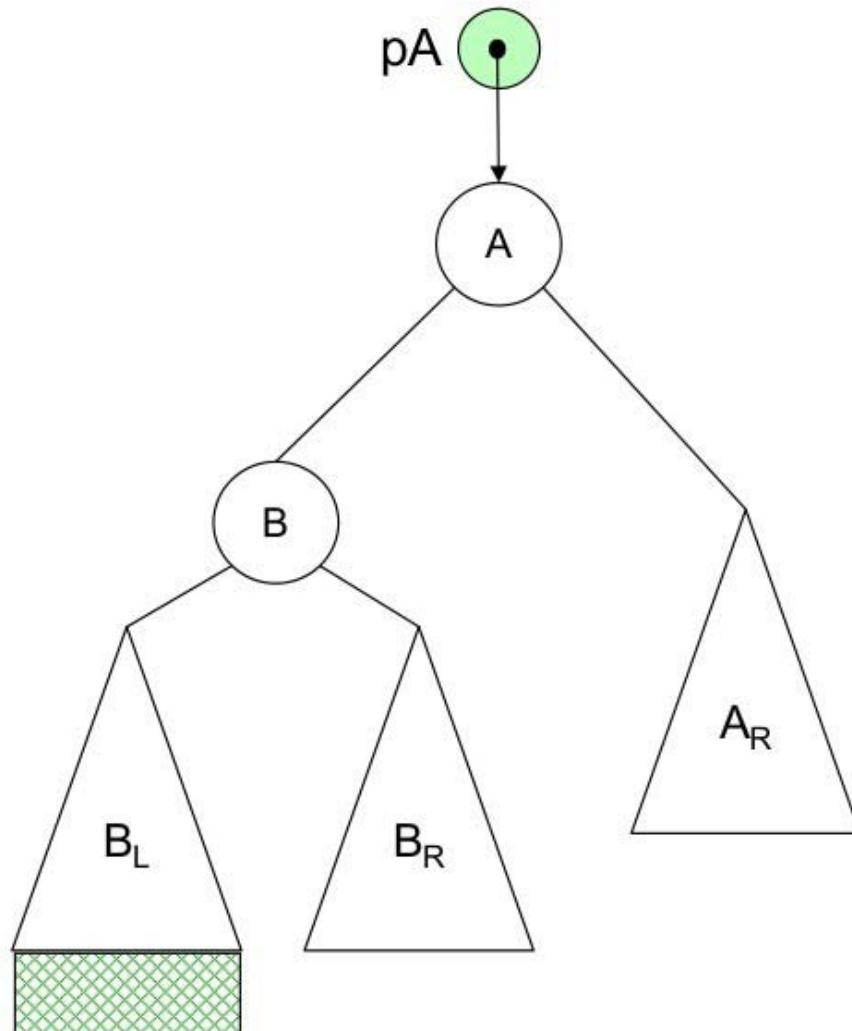


Subárvore desbalanceada após inserção



Altura de B_L aumenta para $h+1$

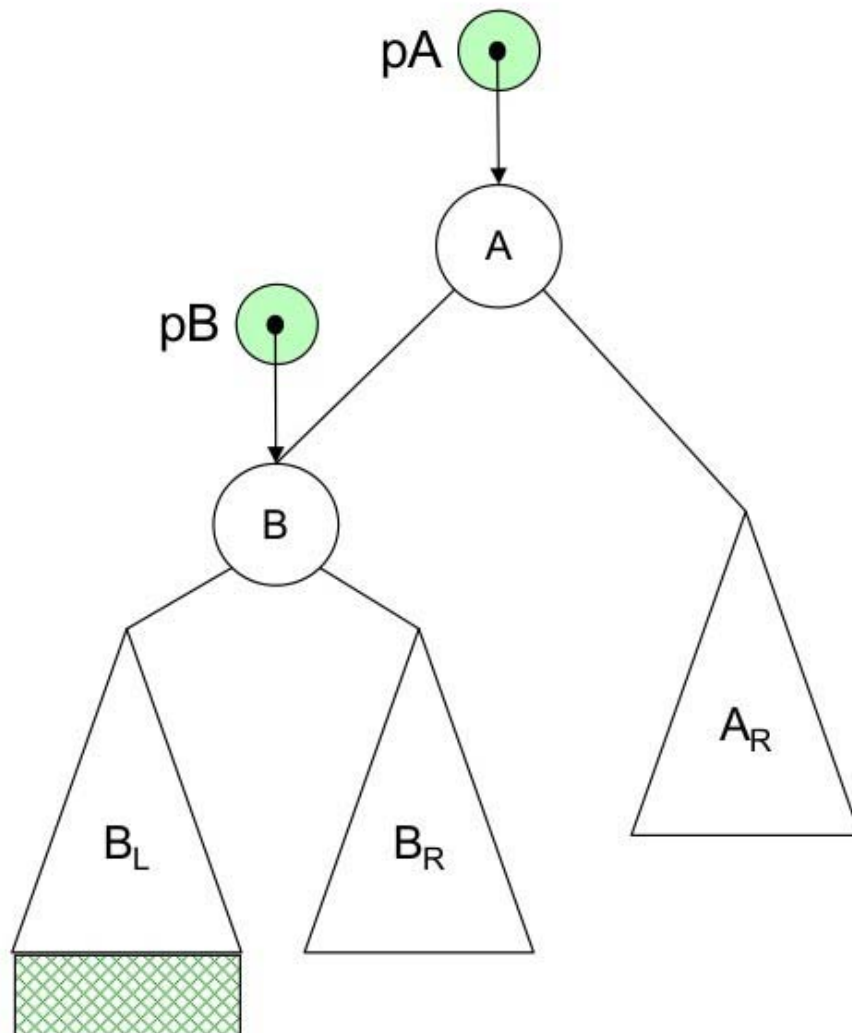
Rotação LL



□ Assumindo pA e pB ponteiros para as subárvores com raízes A e B :

- $pB = pA \rightarrow \text{LeftNode};$
- $pA \rightarrow \text{LeftNode} = pB \rightarrow \text{RightNode};$
- $pB \rightarrow \text{RightNode} = pA;$
- $pA = pB;$

Rotação LL



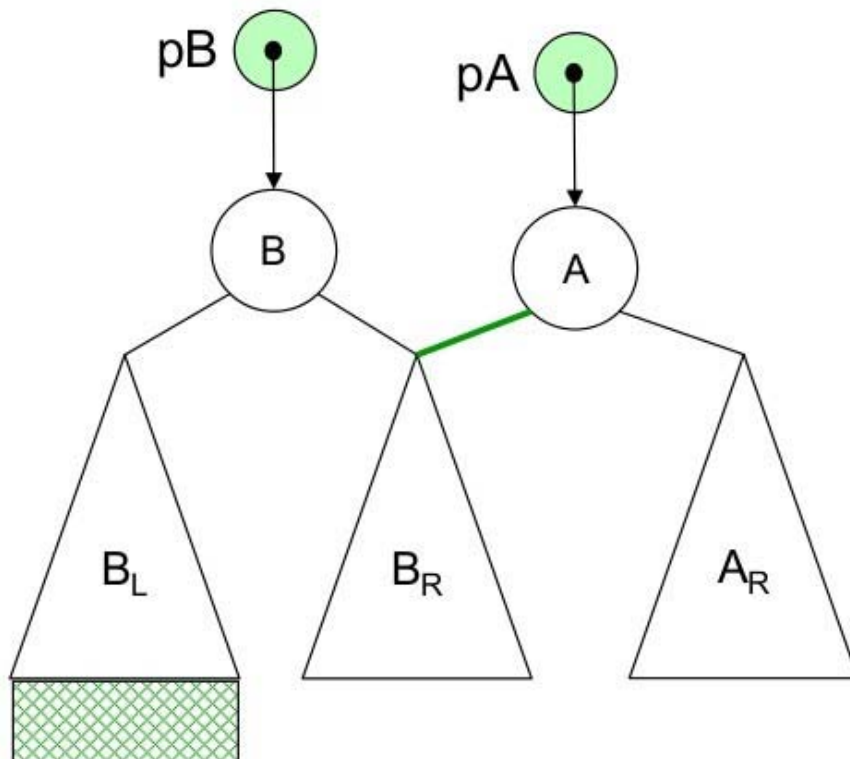
□ Assumindo pA e pB ponteiros para as subárvores com raízes A e B:

- **pB = pA->LeftNode;**
- pA->LeftNode = pB->RightNode;
- pB->RightNode = pA;
- pA = pB;

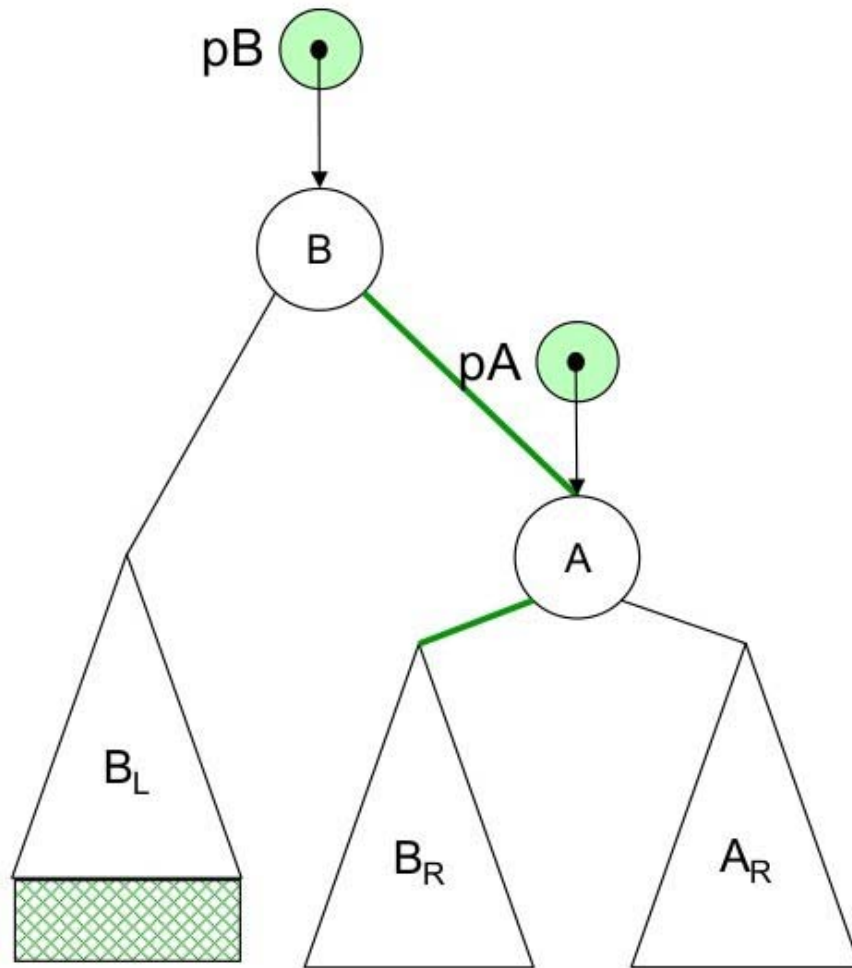
Rotação LL

□ Assumindo pA e pB ponteiros para as subárvores com raízes A e B:

- $pB = pA \rightarrow \text{LeftNode};$
- **$pA \rightarrow \text{LeftNode} = pB \rightarrow \text{RightNode};$**
- $pB \rightarrow \text{RightNode} = pA;$
- $pA = pB;$



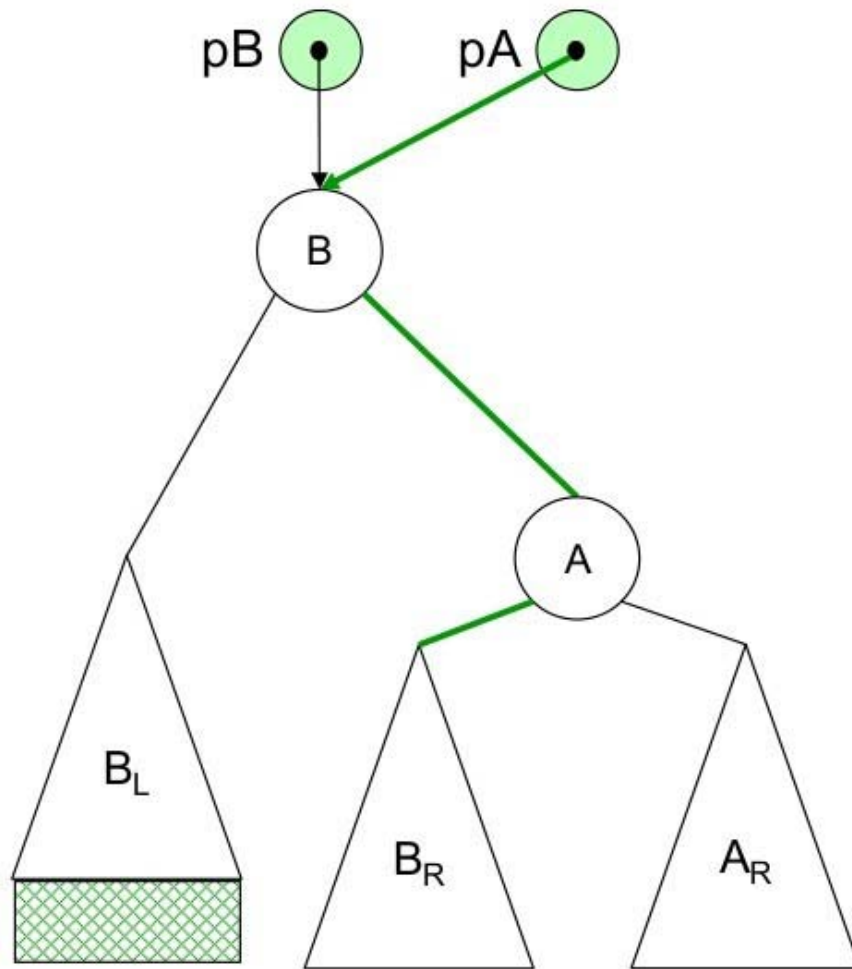
Rotação LL



□ Assumindo pA e pB ponteiros para as subárvores com raízes A e B:

- $pB = pA \rightarrow \text{LeftNode};$
- $pA \rightarrow \text{LeftNode} = pB \rightarrow \text{RightNode};$
- **$pB \rightarrow \text{RightNode} = pA;$**
- $pA = pB;$

Rotação LL

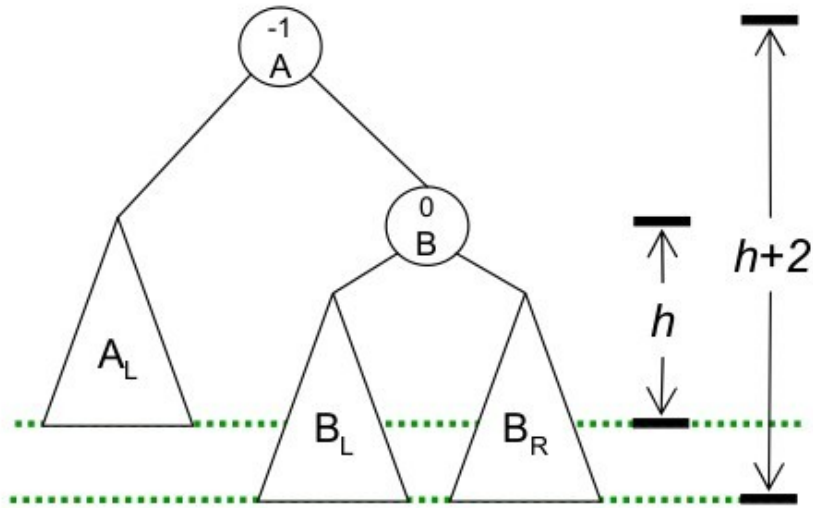


□ Assumindo pA e pB ponteiros para as subárvores com raízes A e B :

- $pB = pA \rightarrow \text{LeftNode};$
- $pA \rightarrow \text{LeftNode} = pB \rightarrow \text{RightNode};$
- $pB \rightarrow \text{RightNode} = pA;$
- **$pA = pB;$**

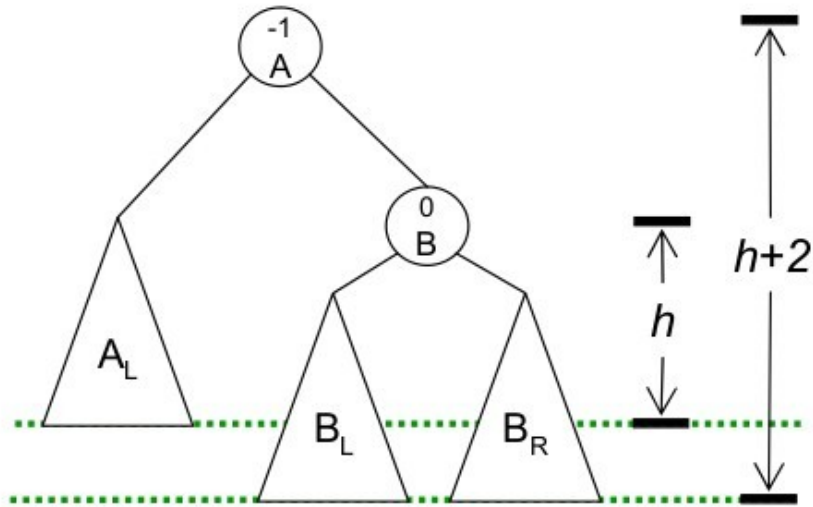
Rotação RR

Subárvore balanceada

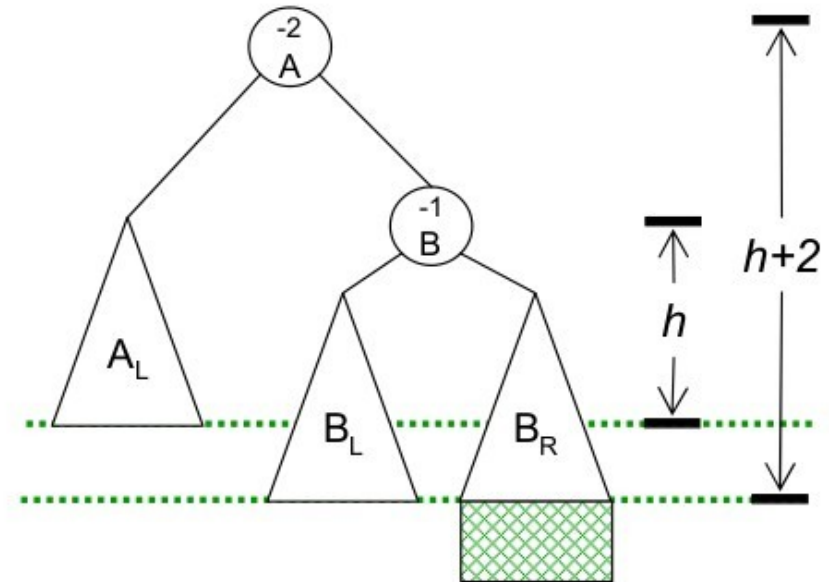


Rotação RR

Subárvore balanceada



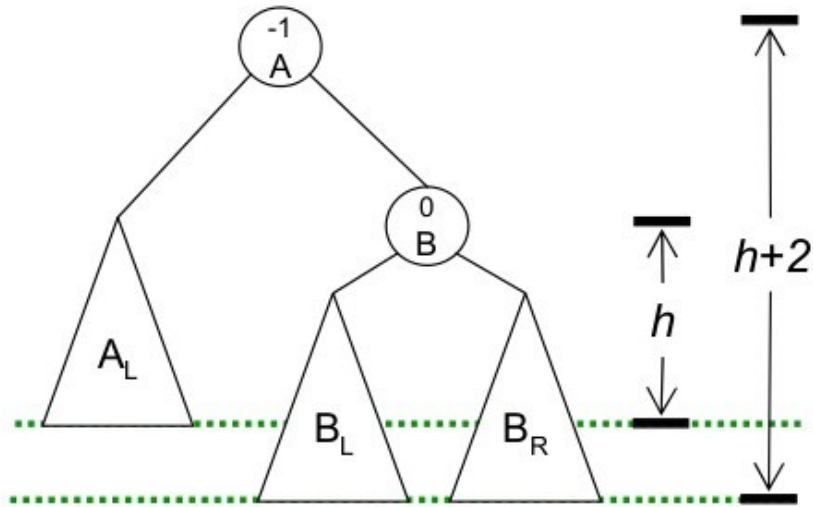
Subárvore desbalanceada após inserção



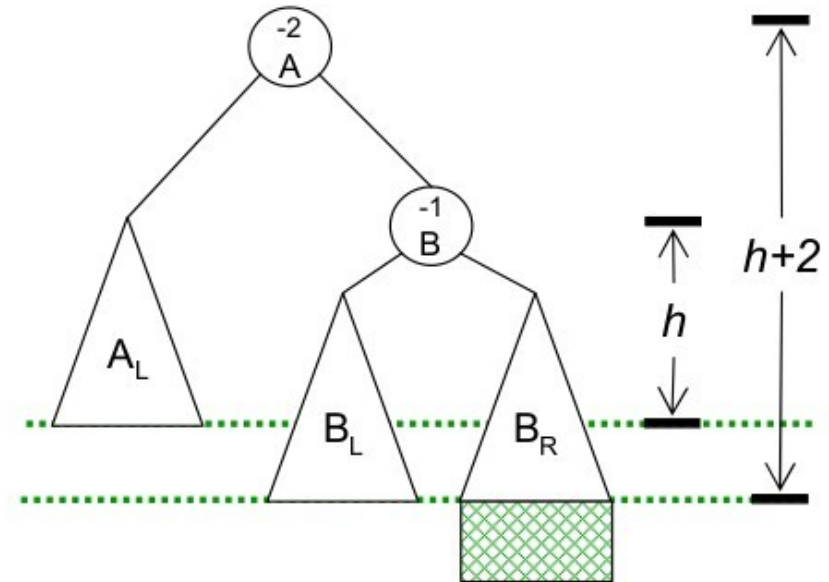
Altura de B_R aumenta para $h+1$

Rotação RR

Subárvore balanceada

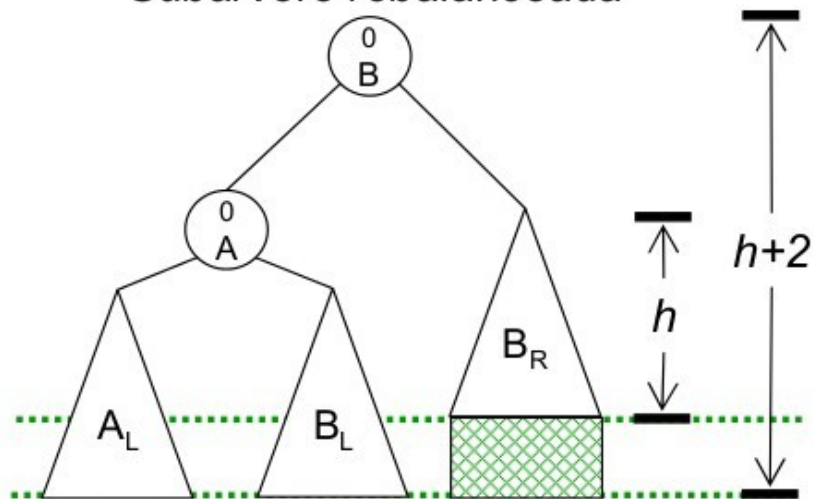


Subárvore desbalanceada após inserção

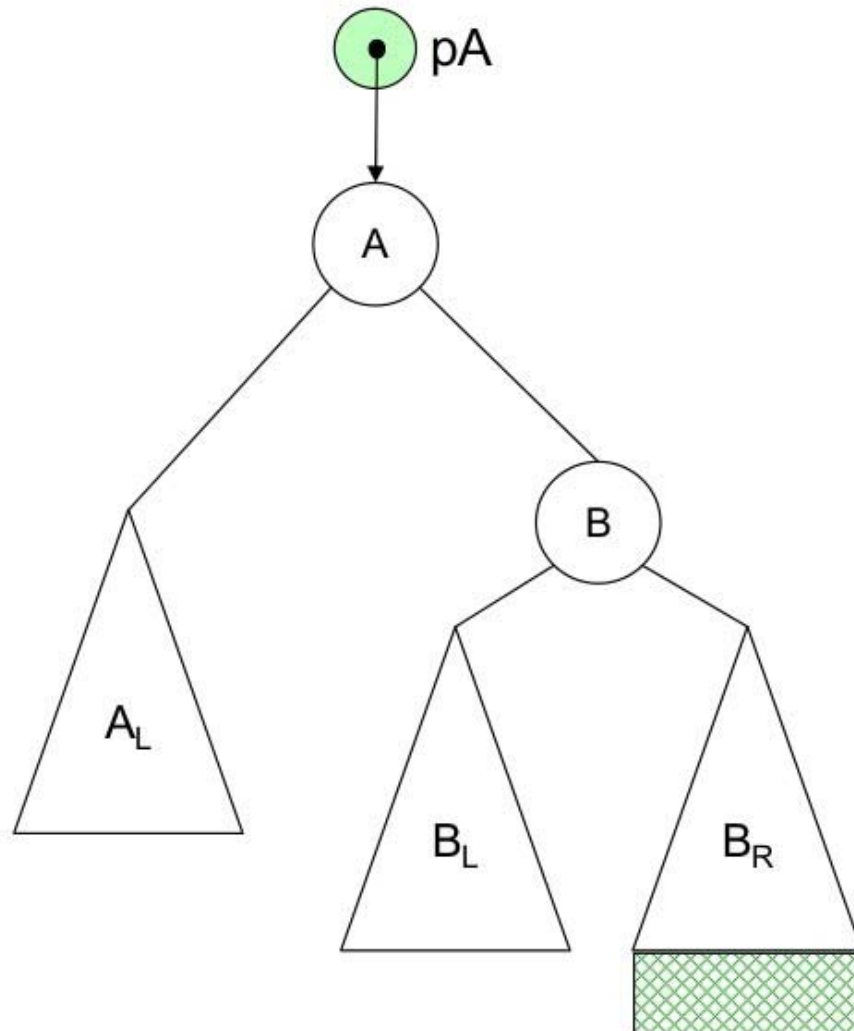


Altura de B_R aumenta para $h+1$

Subárvore rebalanceada



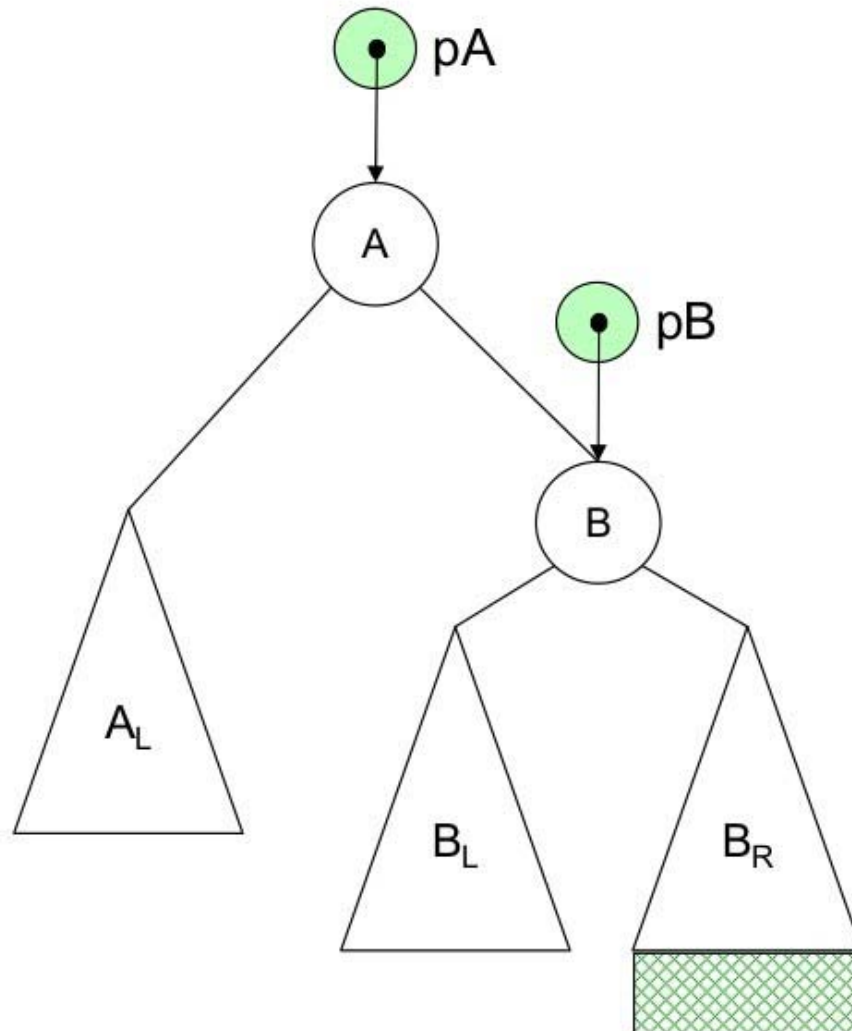
Rotação RR



□ Assumindo pA e pB ponteiros para as subárvores com raízes A e B :

- $pB = pA \rightarrow \text{RightNode}$;
- $pA \rightarrow \text{RightNode} = pB \rightarrow \text{LeftNode}$;
- $pB \rightarrow \text{LeftNode} = pA$;
- $pA = pB$;

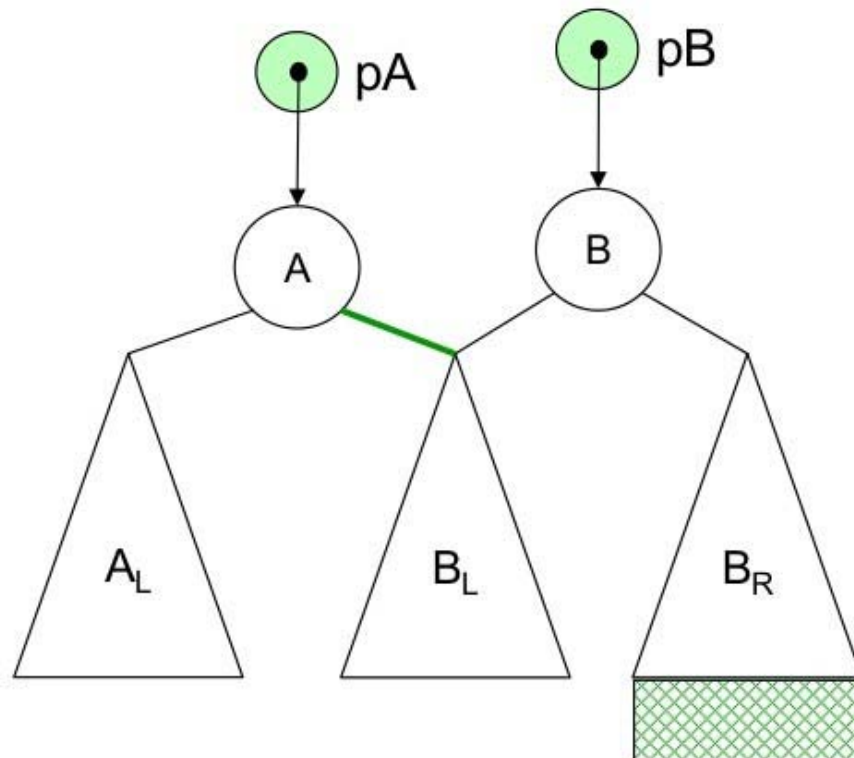
Rotação RR



□ Assumindo pA e pB ponteiros para as subárvores com raízes A e B:

- **pB = pA->RightNode;**
- pA->RightNode = pB->LeftNode;
- pB->LeftNode = pA;
- pA = pB;

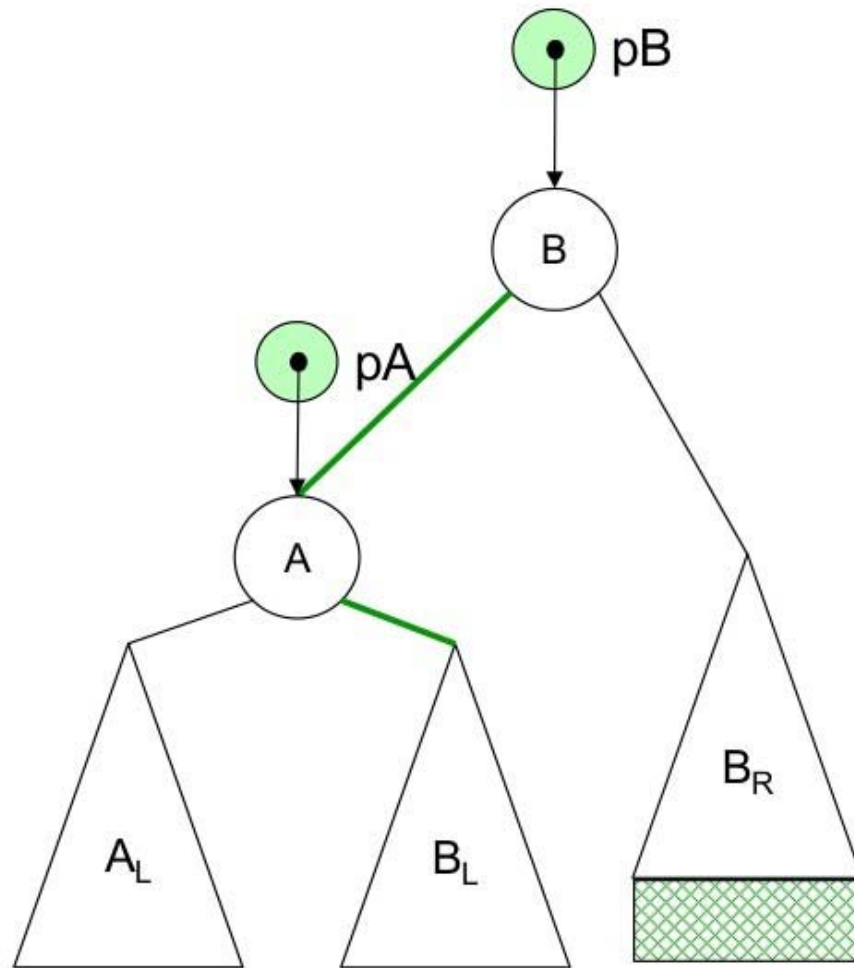
Rotação RR



□ Assumindo pA e pB ponteiros para as subárvores com raízes A e B:

- $pB = pA \rightarrow \text{RightNode};$
- **$pA \rightarrow \text{RightNode} = pB \rightarrow \text{LeftNode};$**
- $pB \rightarrow \text{LeftNode} = pA;$
- $pA = pB;$

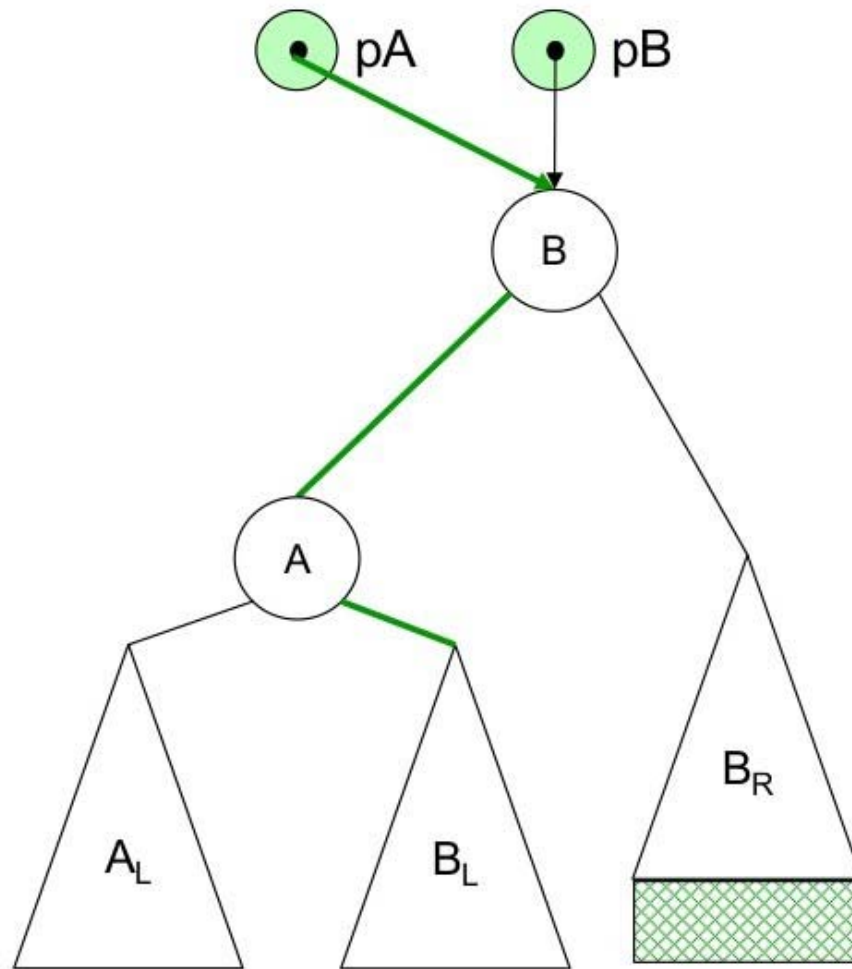
Rotação RR



□ Assumindo pA e pB ponteiros para as subárvores com raízes A e B:

- $pB = pA \rightarrow \text{RightNode};$
- $pA \rightarrow \text{RightNode} = pB \rightarrow \text{LeftNode};$
- **$pB \rightarrow \text{LeftNode} = pA;$**
- $pA = pB;$

Rotação RR

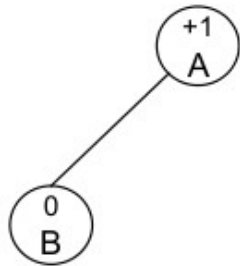


□ Assumindo pA e pB ponteiros para as subárvores com raízes A e B :

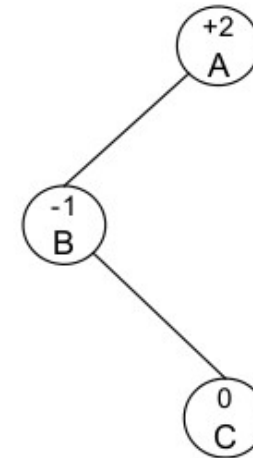
- $pB = pA \rightarrow \text{RightNode};$
- $pA \rightarrow \text{RightNode} = pB \rightarrow \text{LeftNode};$
- $pB \rightarrow \text{LeftNode} = pA;$
- **$pA = pB;$**

Rotação LR (a)

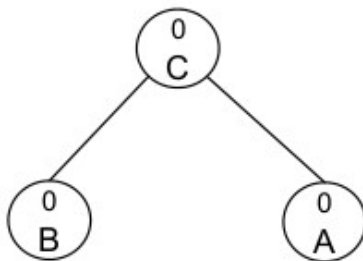
Subárvore balanceada



Subárvore desbalanceada após inserção

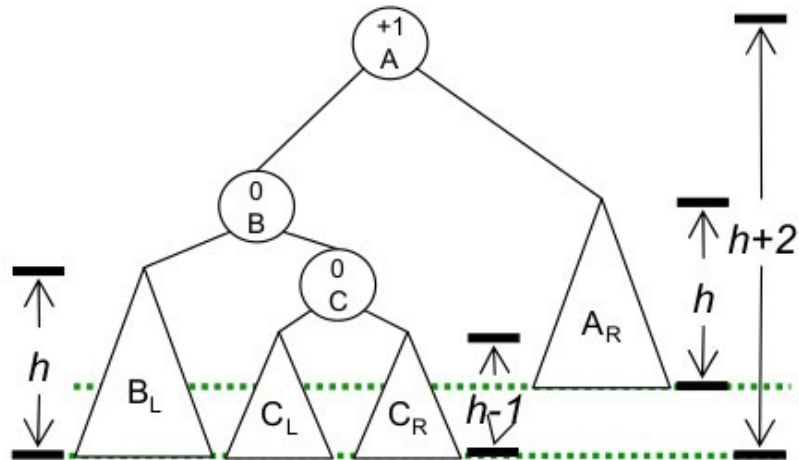


Subárvore rebalanceada



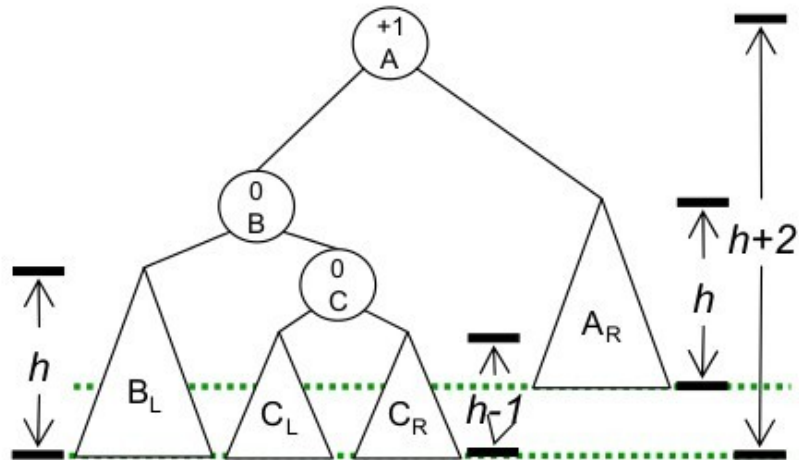
Rotação LR (b)

Subárvore balanceada

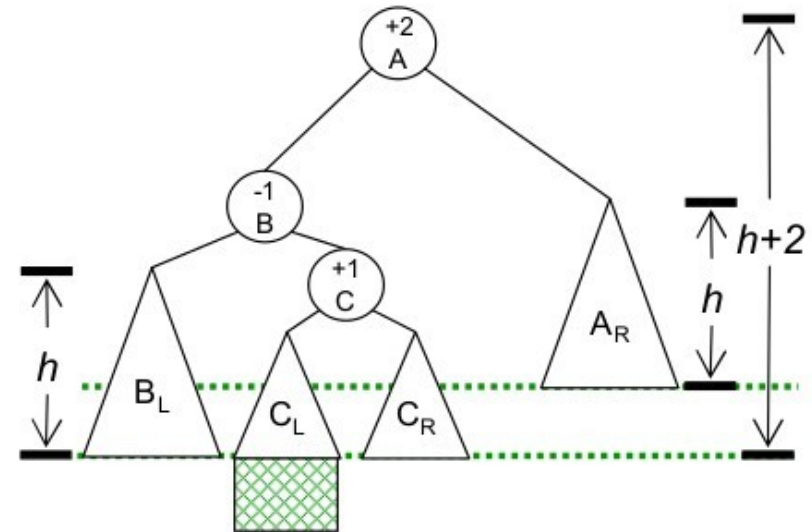


Rotação LR (b)

Subárvore balanceada

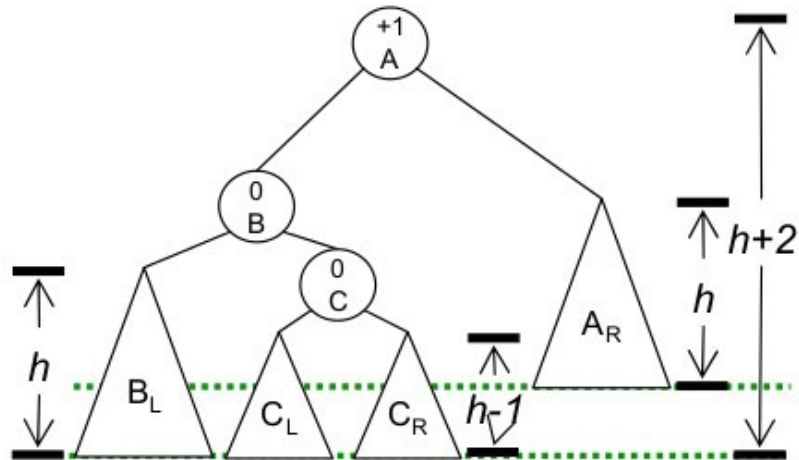


Subárvore desbalanceada após inserção

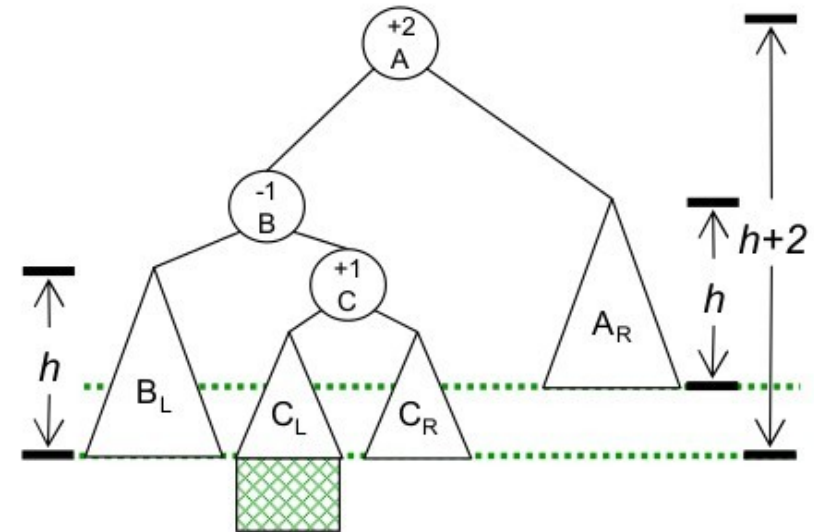


Rotação LR (b)

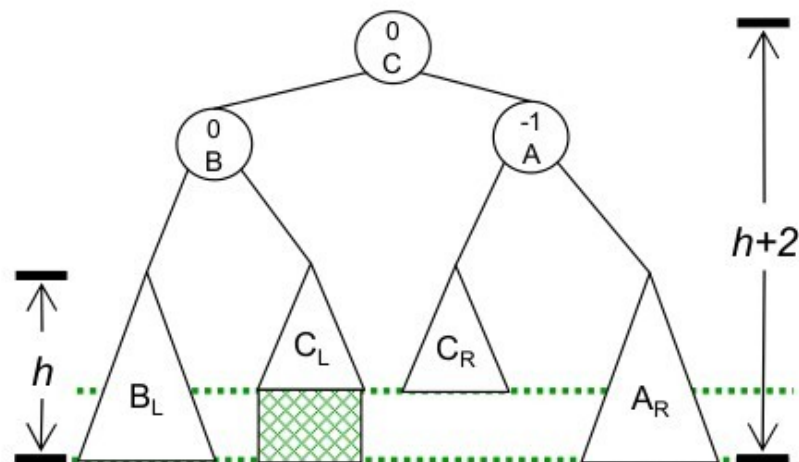
Subárvore balanceada



Subárvore desbalanceada após inserção

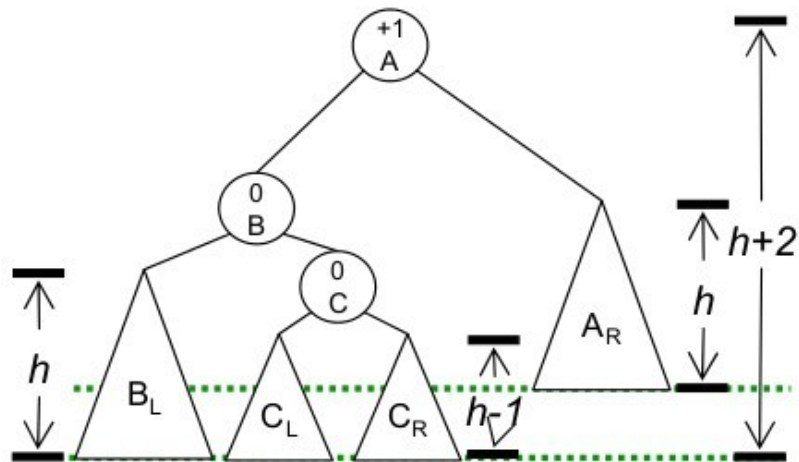


Subárvore rebalanceada



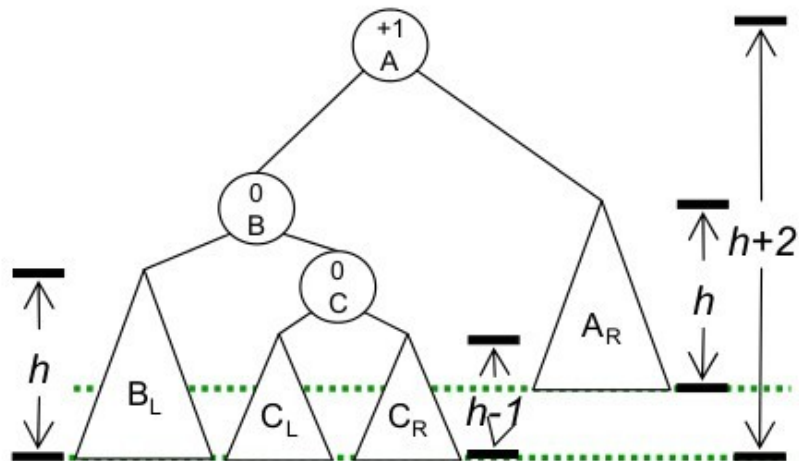
Rotação LR (c)

Subárvore balanceada

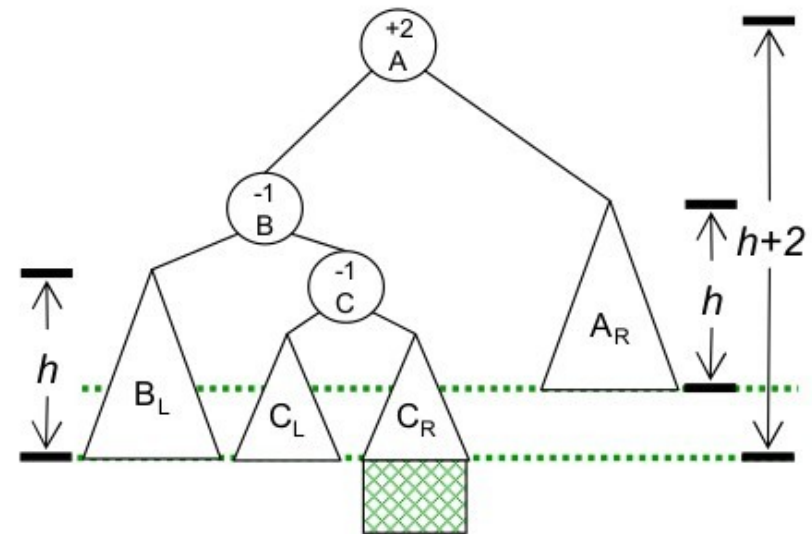


Rotação LR (c)

Subárvore balanceada

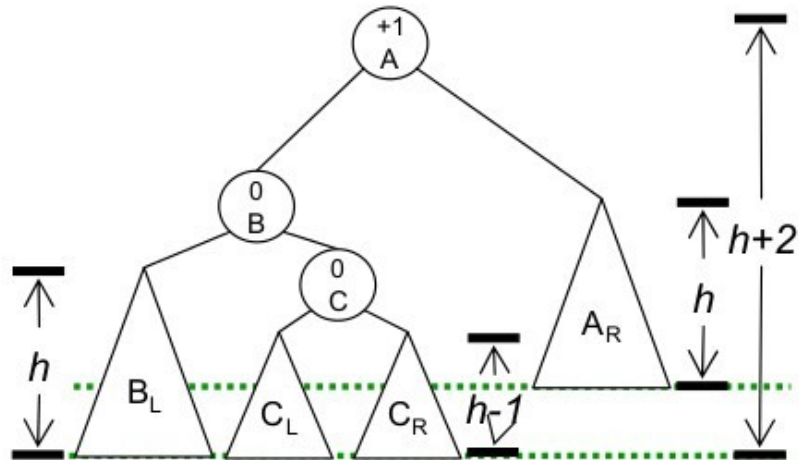


Subárvore desbalanceada após inserção

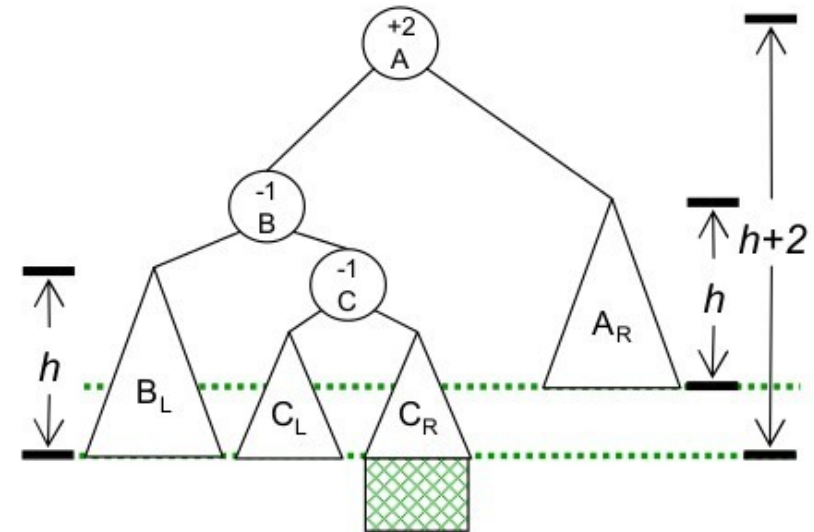


Rotação LR (c)

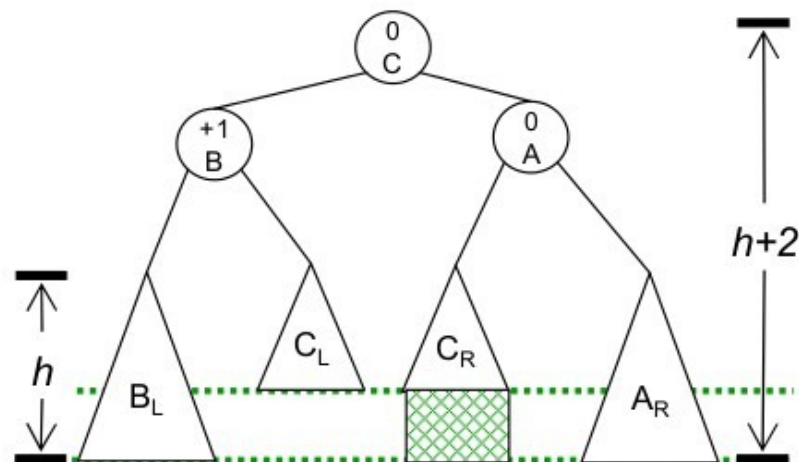
Subárvore balanceada



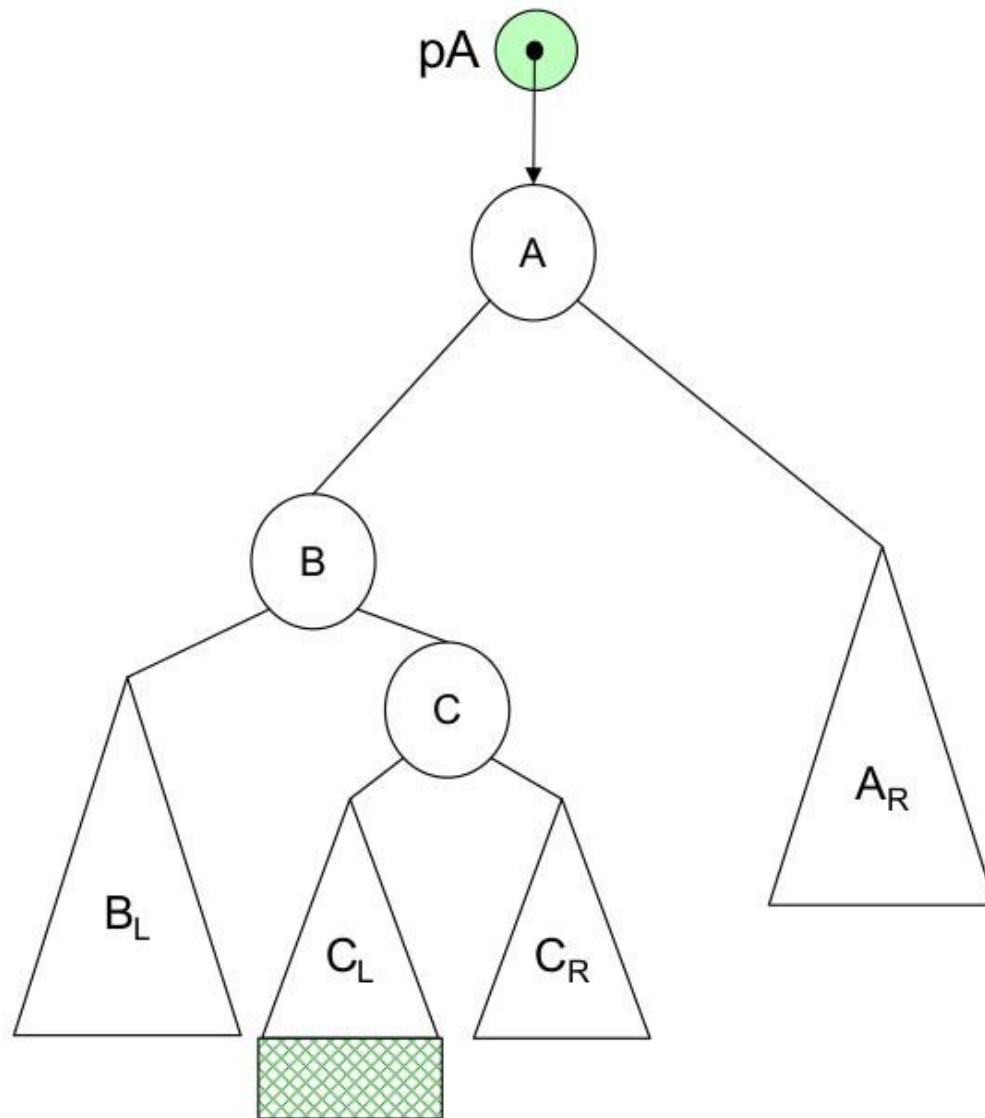
Subárvore desbalanceada após inserção



Subárvore rebalanceada



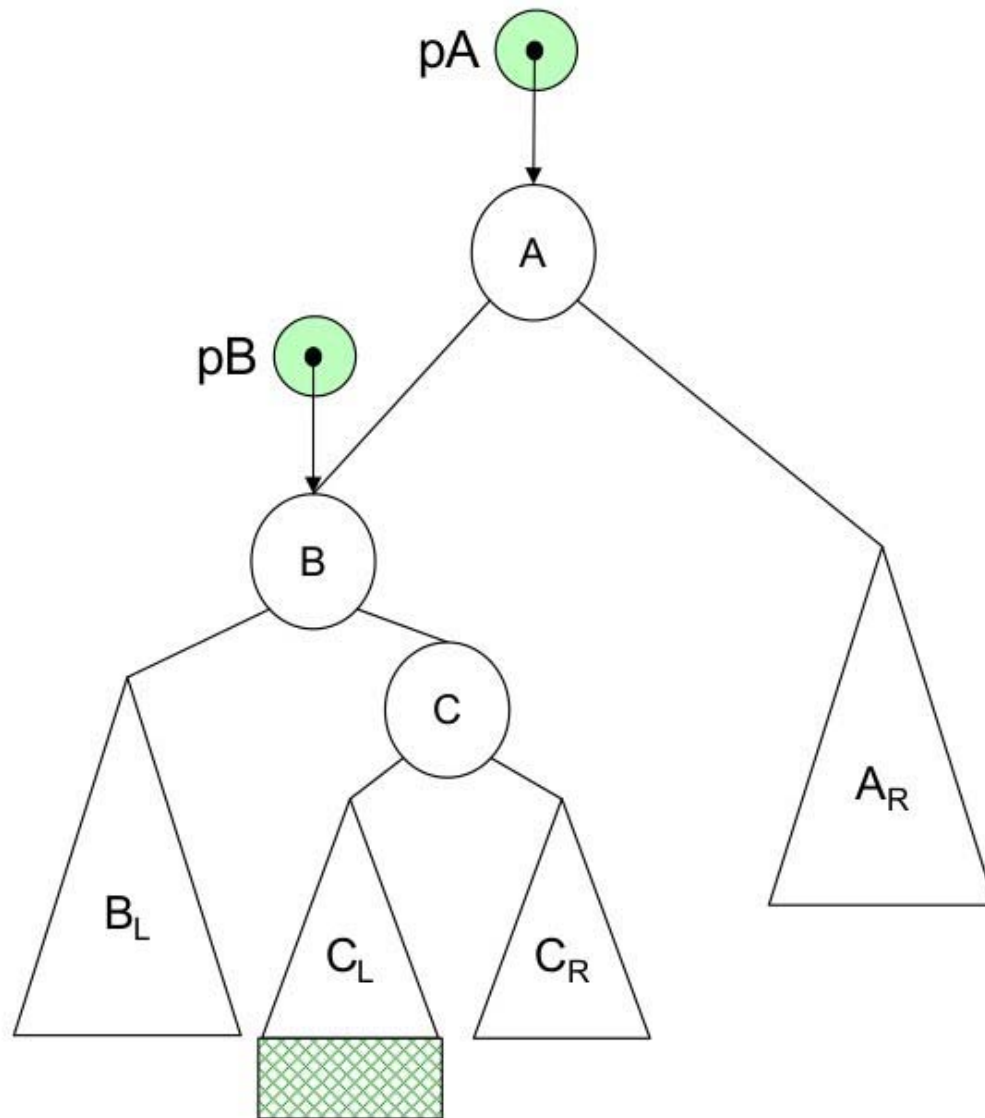
Rotação LR



Assumindo pA , pB e pC ponteiros para as subárvores com raízes A , B e C :

- $pB = pA \rightarrow \text{LeftNode};$
- $pC = pB \rightarrow \text{RightNode};$
- $pB \rightarrow \text{RightNode} = pC \rightarrow \text{LeftNode};$
- $pC \rightarrow \text{LeftNode} = pB;$
- $pA \rightarrow \text{LeftNode} = pC \rightarrow \text{RightNode};$
- $pC \rightarrow \text{RightNode} = pA;$
- $pA = pC;$

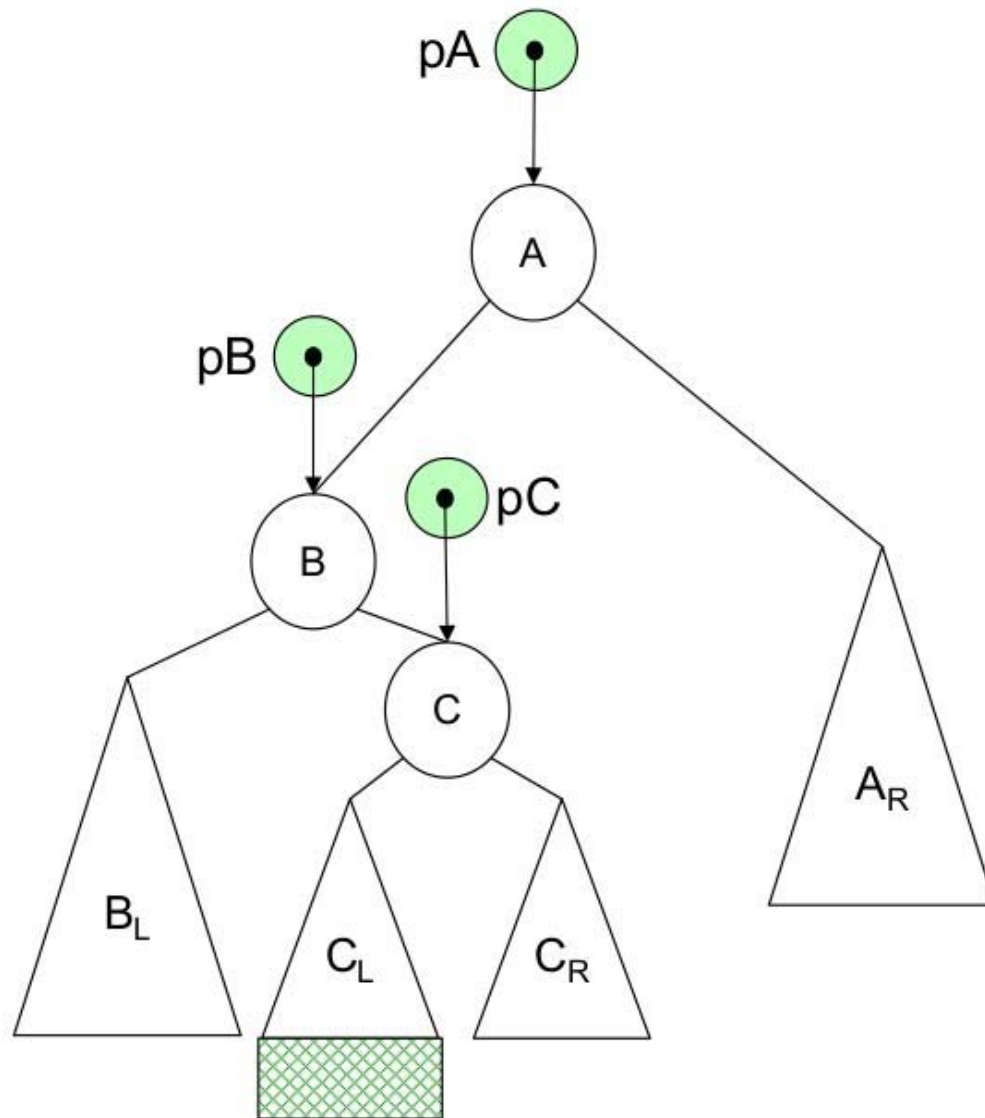
Rotação LR



Assumindo pA, pB e pC ponteiros para as subárvores com raízes A, B e C:

- **pB = pA->LeftNode;**
- pC = pB->RightNode;
- pB->RightNode = pC->LeftNode;
- pC->LeftNode = pB;
- pA->LeftNode = pC->RightNode;
- pC->RightNode = pA;
- pA = pC;

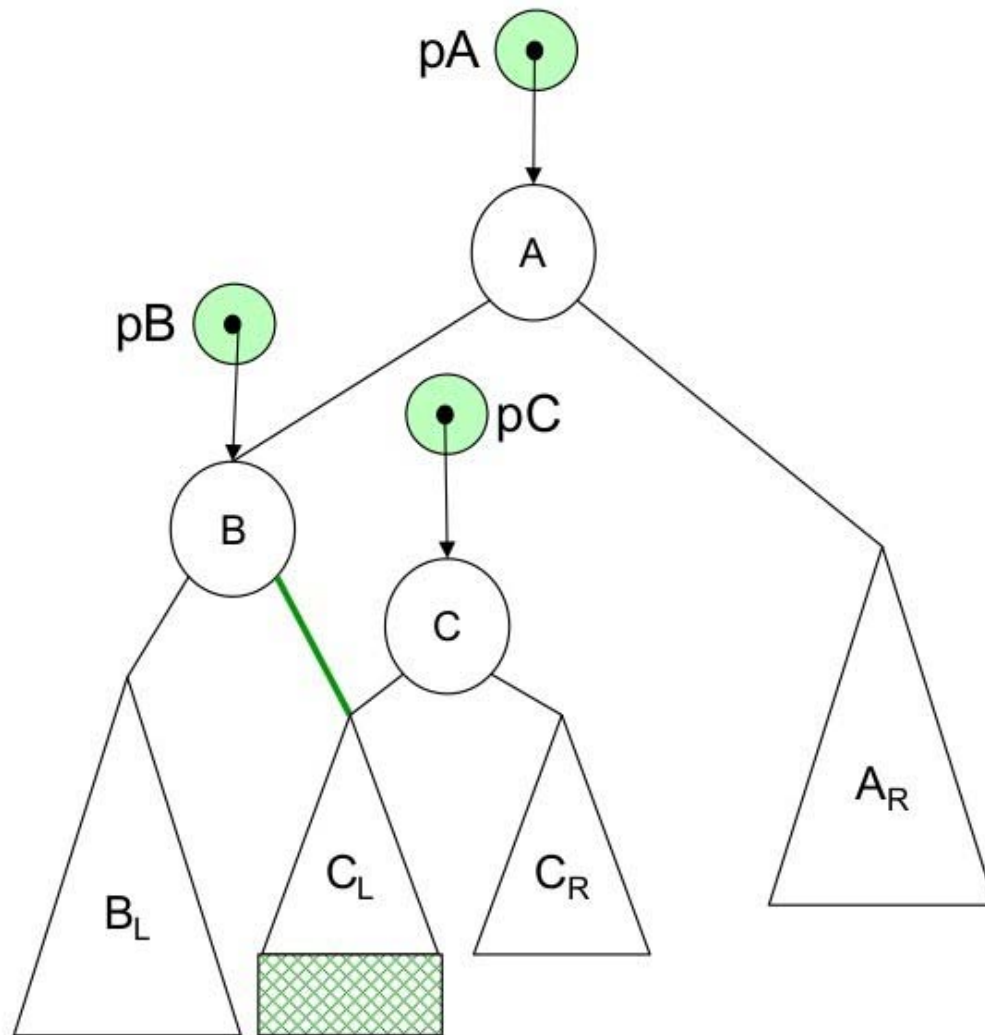
Rotação LR



□ Assumindo pA, pB e pC ponteiros para as subárvores com raízes A, B e C:

- $pB = pA \rightarrow \text{LeftNode};$
- **$pC = pB \rightarrow \text{RightNode};$**
- $pB \rightarrow \text{RightNode} = pC \rightarrow \text{LeftNode};$
- $pC \rightarrow \text{LeftNode} = pB;$
- $pA \rightarrow \text{LeftNode} = pC \rightarrow \text{RightNode};$
- $pC \rightarrow \text{RightNode} = pA;$
- $pA = pC;$

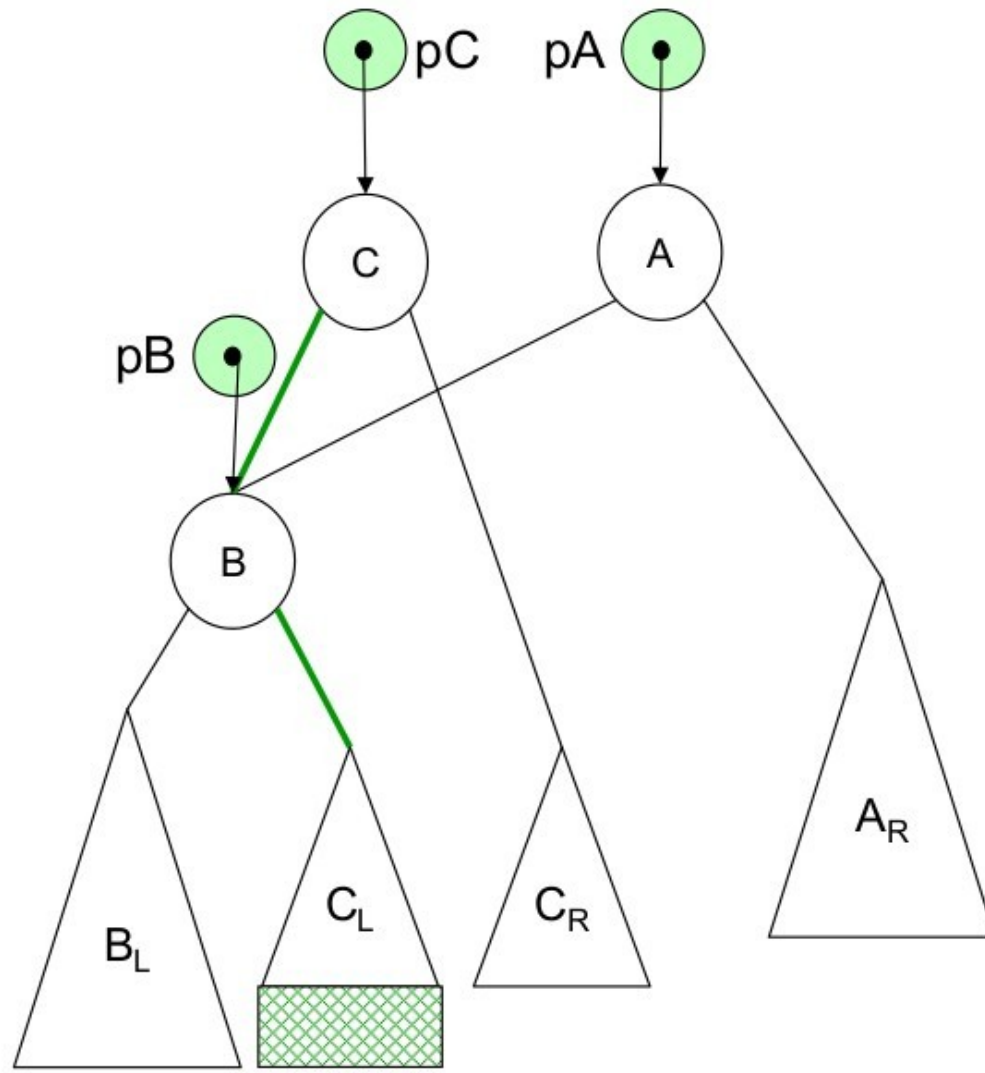
Rotação LR



Assumindo pA, pB e pC ponteiros para as subárvores com raízes A, B e C:

- $pB = pA \rightarrow \text{LeftNode};$
- $pC = pB \rightarrow \text{RightNode};$
- **$pB \rightarrow \text{RightNode} = pC \rightarrow \text{LeftNode};$**
- $pC \rightarrow \text{LeftNode} = pB;$
- $pA \rightarrow \text{LeftNode} = pC \rightarrow \text{RightNode};$
- $pC \rightarrow \text{RightNode} = pA;$
- $pA = pC;$

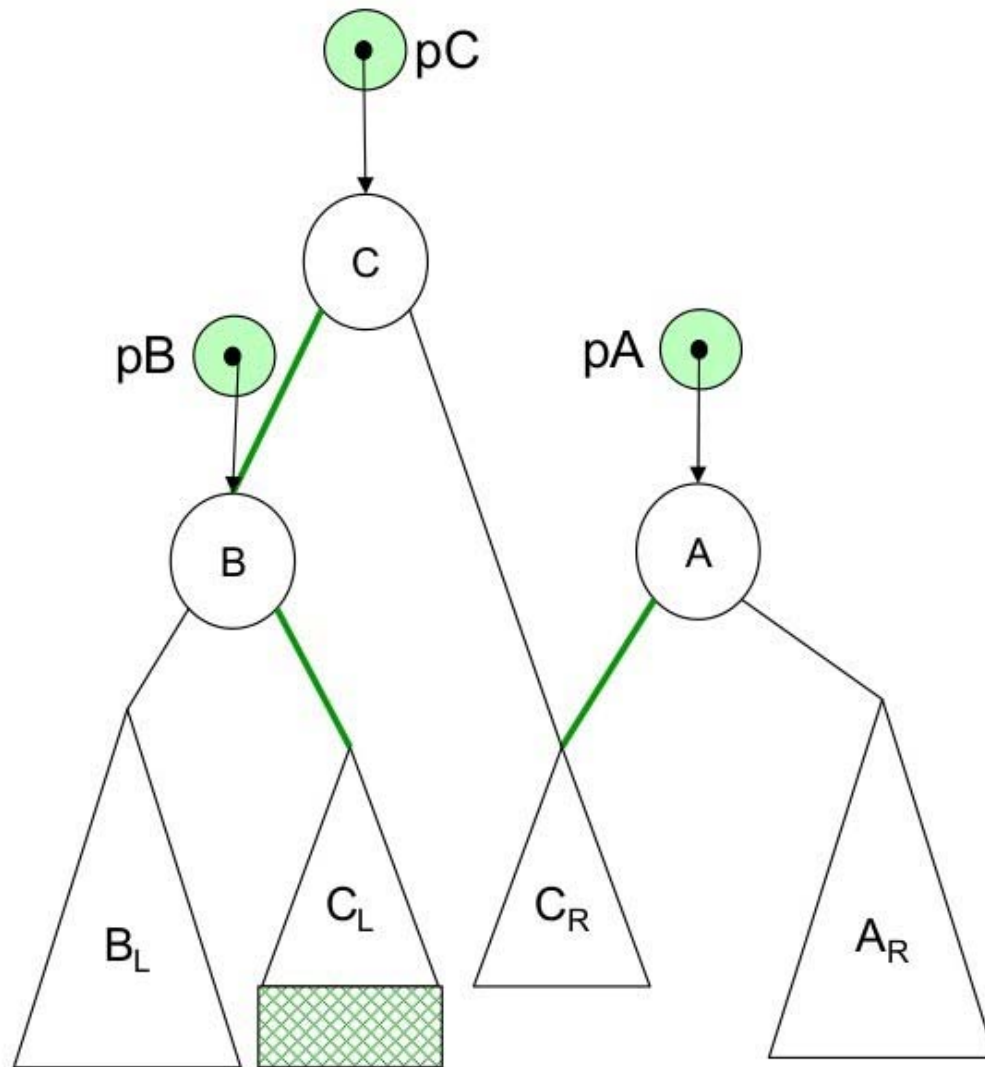
Rotação LR



□ Assumindo pA, pB e pC ponteiros para as subárvores com raízes A, B e C:

- $pB = pA \rightarrow \text{LeftNode};$
- $pC = pB \rightarrow \text{RightNode};$
- $pB \rightarrow \text{RightNode} = pC \rightarrow \text{LeftNode};$
- **$pC \rightarrow \text{LeftNode} = pB;$**
- $pA \rightarrow \text{LeftNode} = pC \rightarrow \text{RightNode};$
- $pC \rightarrow \text{RightNode} = pA;$
- $pA = pC;$

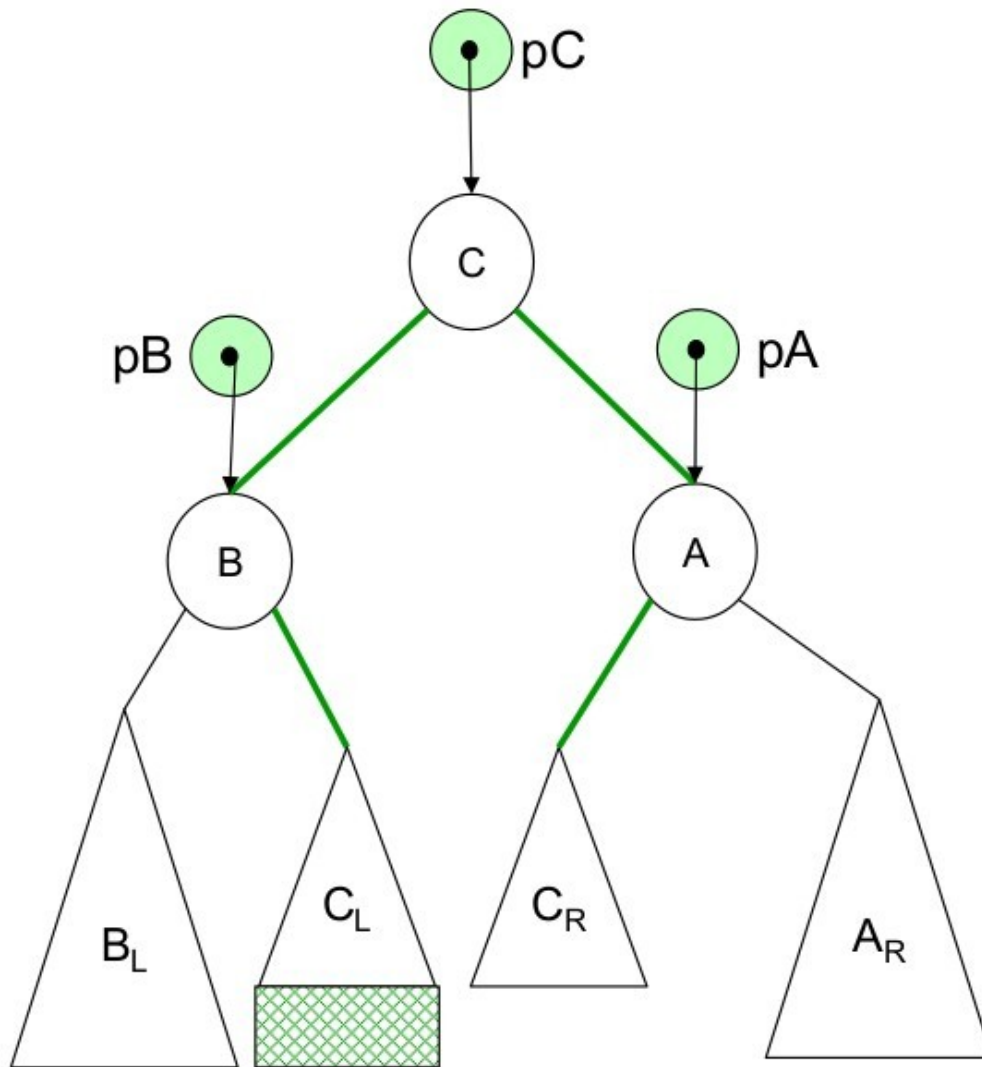
Rotação LR



□ Assumindo pA, pB e pC ponteiros para as subárvores com raízes A, B e C:

- $pB = pA \rightarrow \text{LeftNode};$
- $pC = pB \rightarrow \text{RightNode};$
- $pB \rightarrow \text{RightNode} = pC \rightarrow \text{LeftNode};$
- $pC \rightarrow \text{LeftNode} = pB;$
- **$pA \rightarrow \text{LeftNode} = pC \rightarrow \text{RightNode};$**
- $pC \rightarrow \text{RightNode} = pA;$
- $pA = pC;$

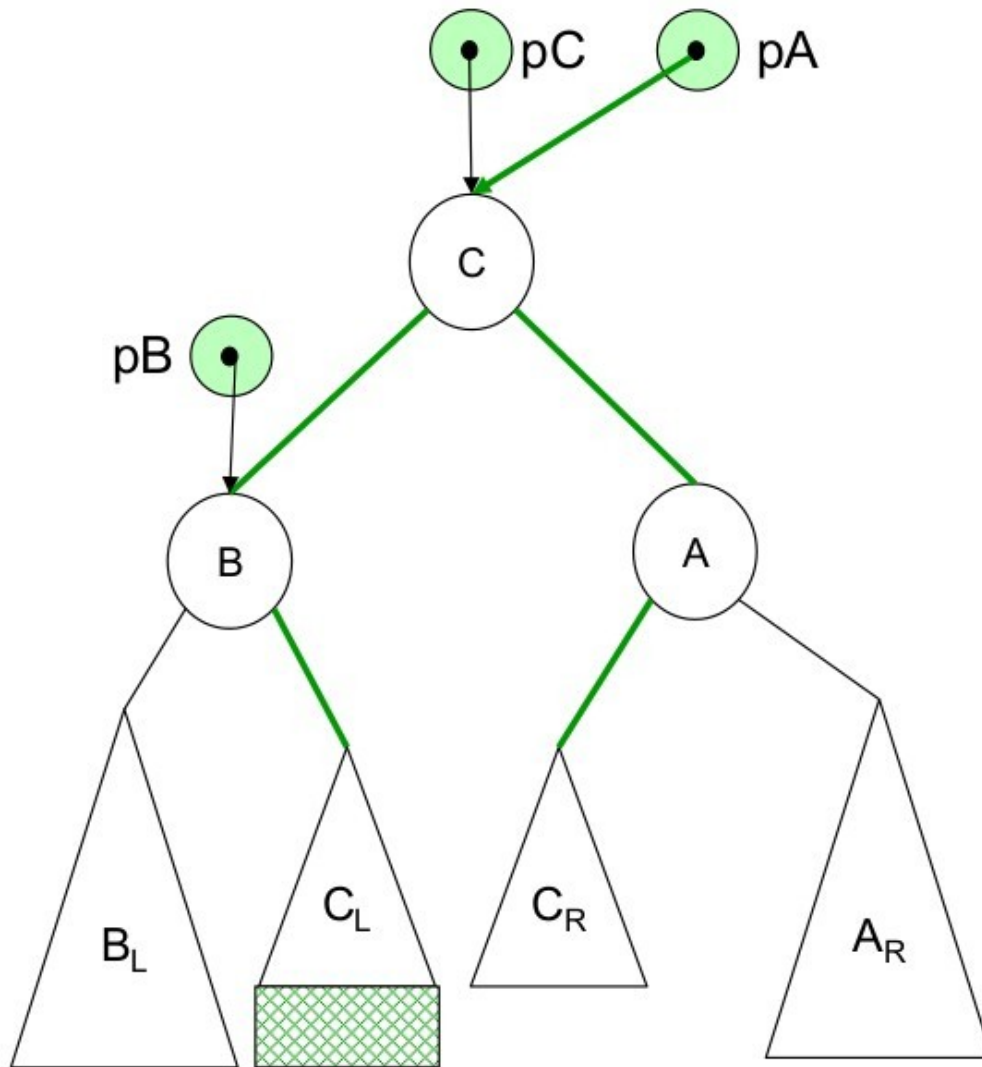
Rotação LR



Assumindo pA, pB e pC ponteiros para as subárvores com raízes A, B e C:

- $pB = pA \rightarrow \text{LeftNode};$
- $pC = pB \rightarrow \text{RightNode};$
- $pB \rightarrow \text{RightNode} = pC \rightarrow \text{LeftNode};$
- $pC \rightarrow \text{LeftNode} = pB;$
- $pA \rightarrow \text{LeftNode} = pC \rightarrow \text{RightNode};$
- **$pC \rightarrow \text{RightNode} = pA;$**
- $pA = pC;$

Rotação LR

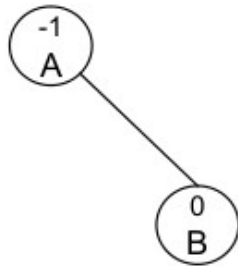


□ Assumindo pA, pB e pC ponteiros para as subárvores com raízes A, B e C:

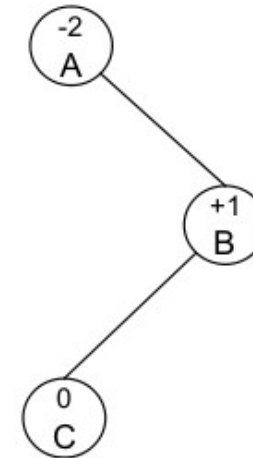
- $pB = pA \rightarrow \text{LeftNode};$
- $pC = pB \rightarrow \text{RightNode};$
- $pB \rightarrow \text{RightNode} = pC \rightarrow \text{LeftNode};$
- $pC \rightarrow \text{LeftNode} = pB;$
- $pA \rightarrow \text{LeftNode} = pC \rightarrow \text{RightNode};$
- $pC \rightarrow \text{RightNode} = pA;$
- **$pA = pC;$**

Rotação RL (a)

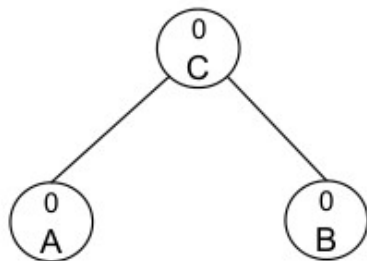
Subárvore balanceada



Subárvore desbalanceada após inserção

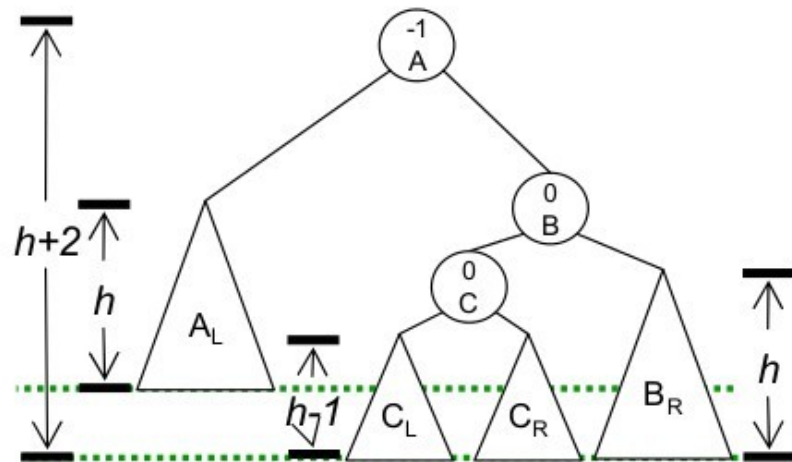


Subárvore rebalanceada



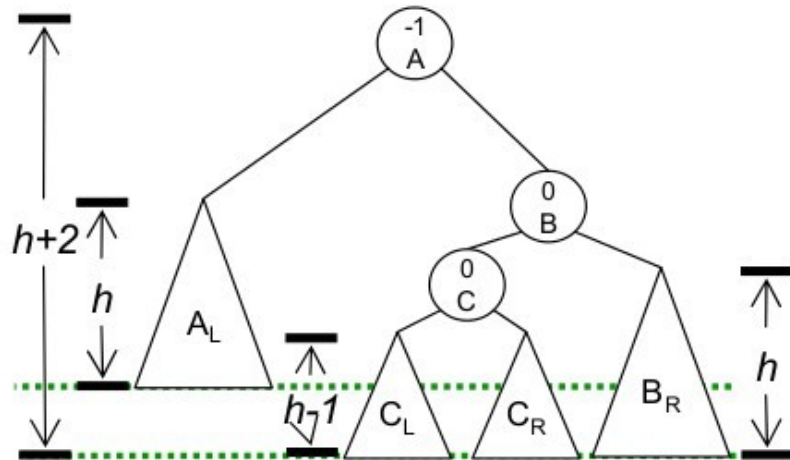
Rotação RL (b)

Subárvore balanceada

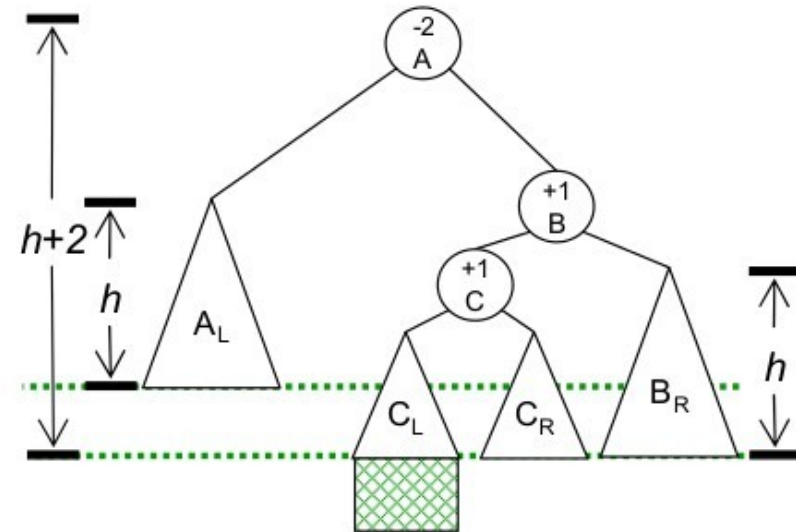


Rotação RL (b)

Subárvore balanceada

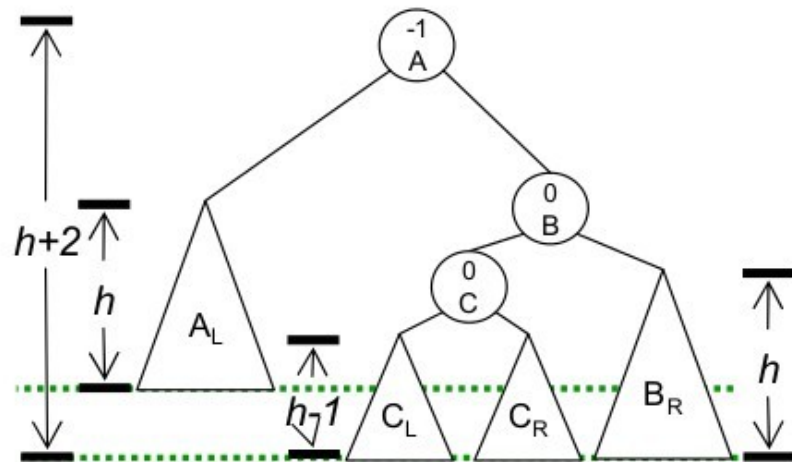


Subárvore desbalanceada após inserção

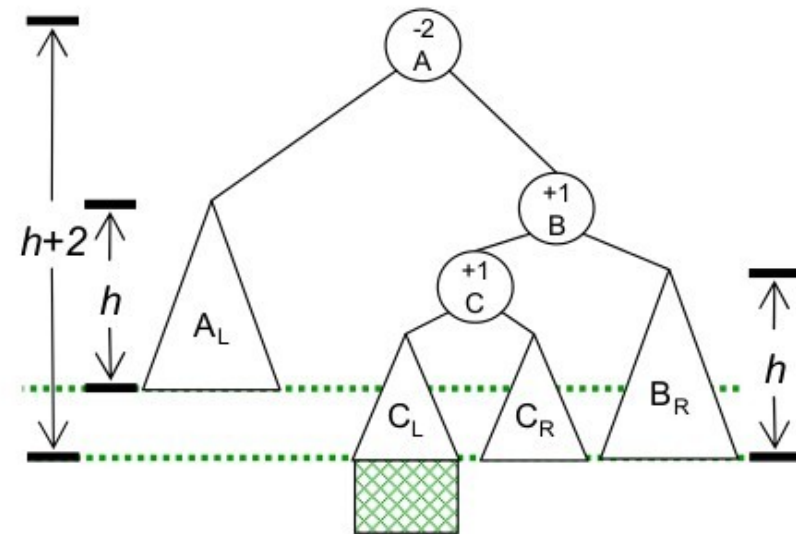


Rotação RL (b)

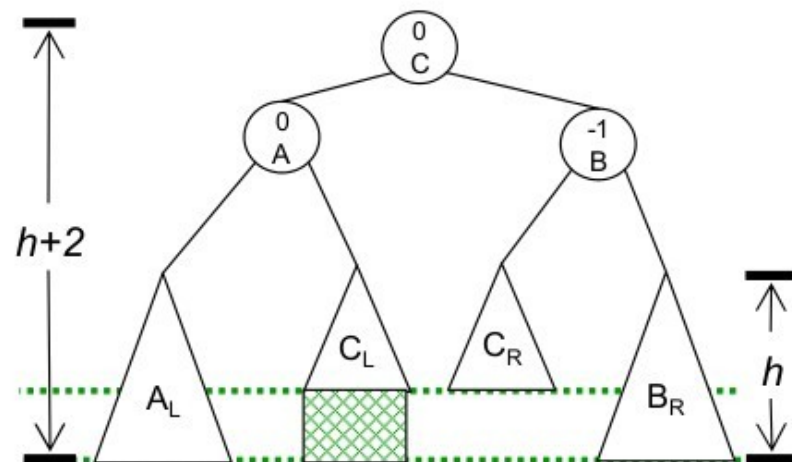
Subárvore balanceada



Subárvore desbalanceada após inserção

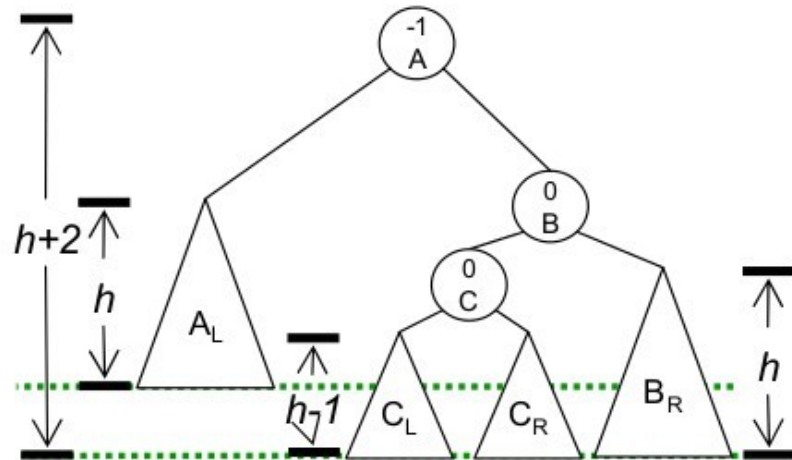


Subárvore rebalanceada

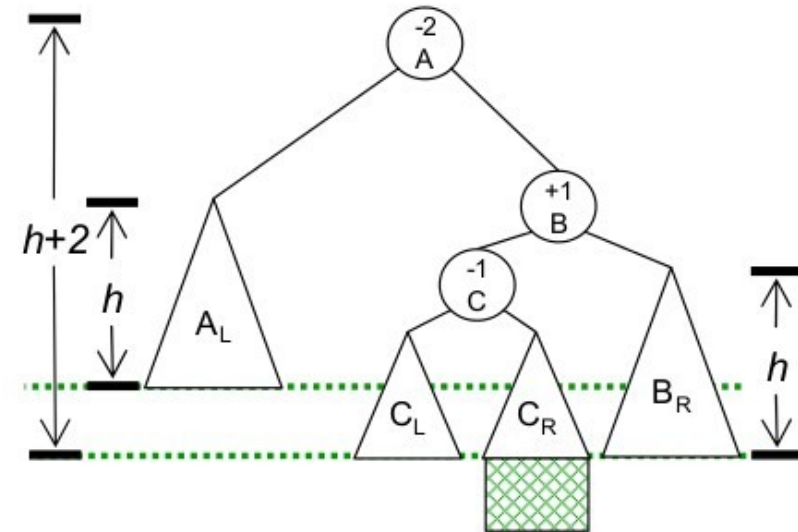


Rotação RL (c)

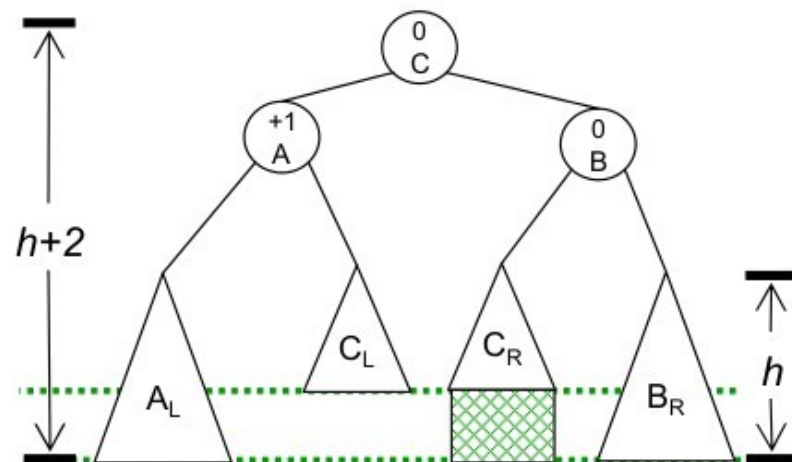
Subárvore balanceada



Subárvore desbalanceada após inserção

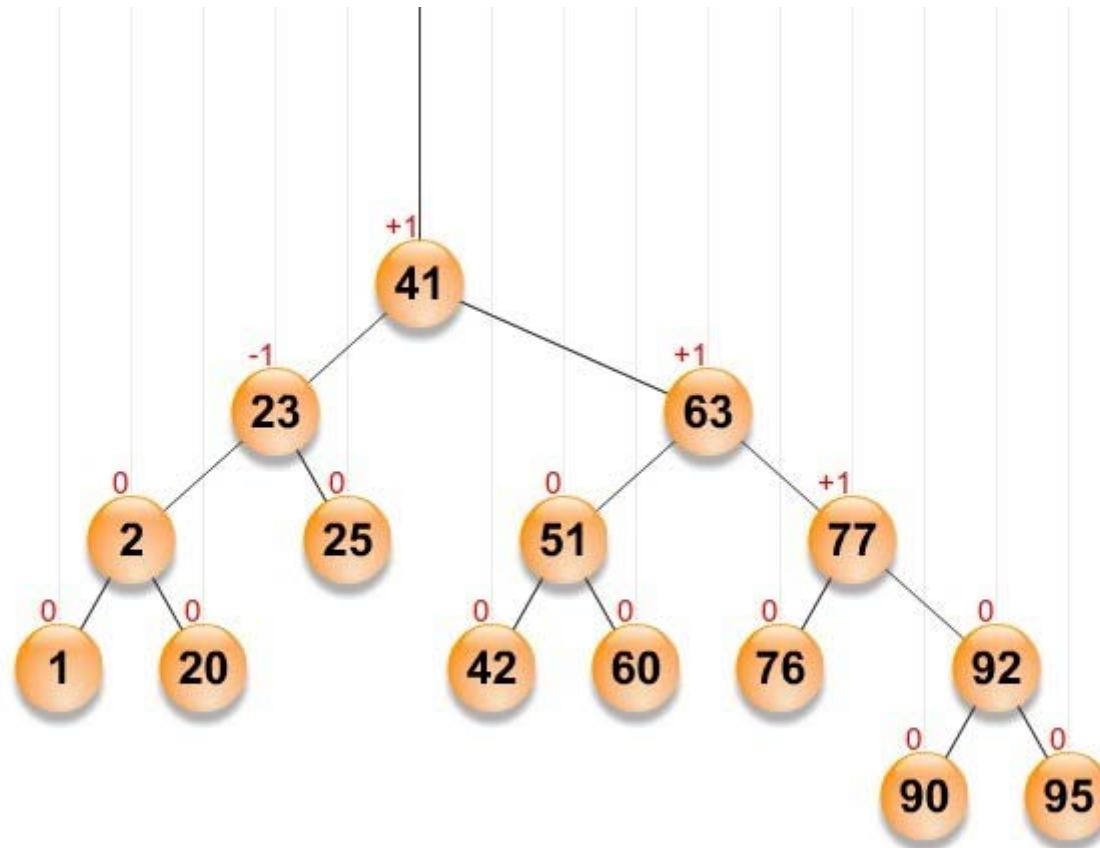


Subárvore rebalanceada



Ferramenta de visualização: AVL

<http://www.qmatica.com/DataStructures/Trees/AVL/AVLTree.html>

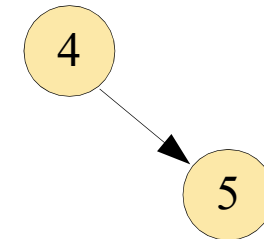


$$-1 * (h_L - h_R)$$

Atividade em aula

- Suponha que serão realizadas as seguintes inserções de chaves na árvore AVL ao lado:

- 7
- 2
- 1
- 3
- 6



- (a) Apresente a árvore AVL resultante (1 árvore)
- (b) Indique o tipo de rotações consideradas em cada inserção

Atividade em aula

Operações

- 7 : RR
- 2 : --
- 1 : LL
- 3 : LR
- 6 : RL

