AULA 19
ESTRUTURA DE DADOS

Árvores AVL: Complexidade e Remoção

Árvore AVL – Rotações - Relembrando

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.

Árvore AVL – Rotações - Relembrando

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

LL: Y inserido na subárvore esquerda da subárvore esquerda de A

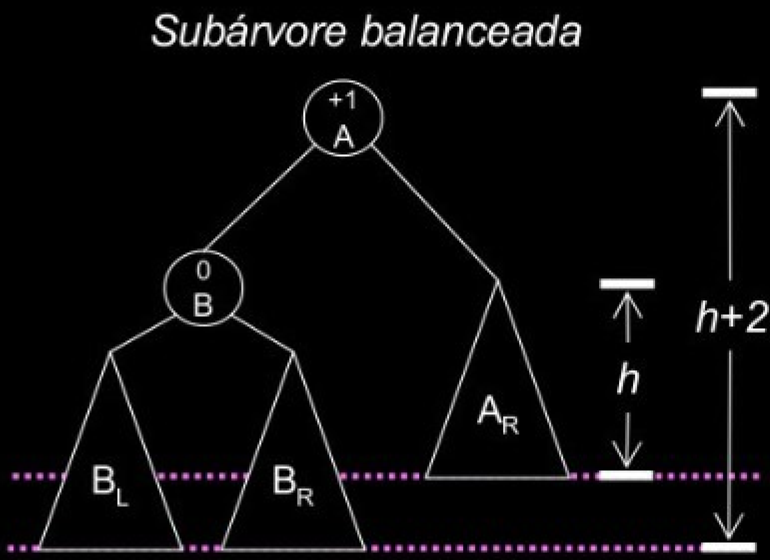
LR: Y inserido na subárvore direita da subárvore esquerda de A

RR: Y inserido na subárvore direita da subárvore direita de A

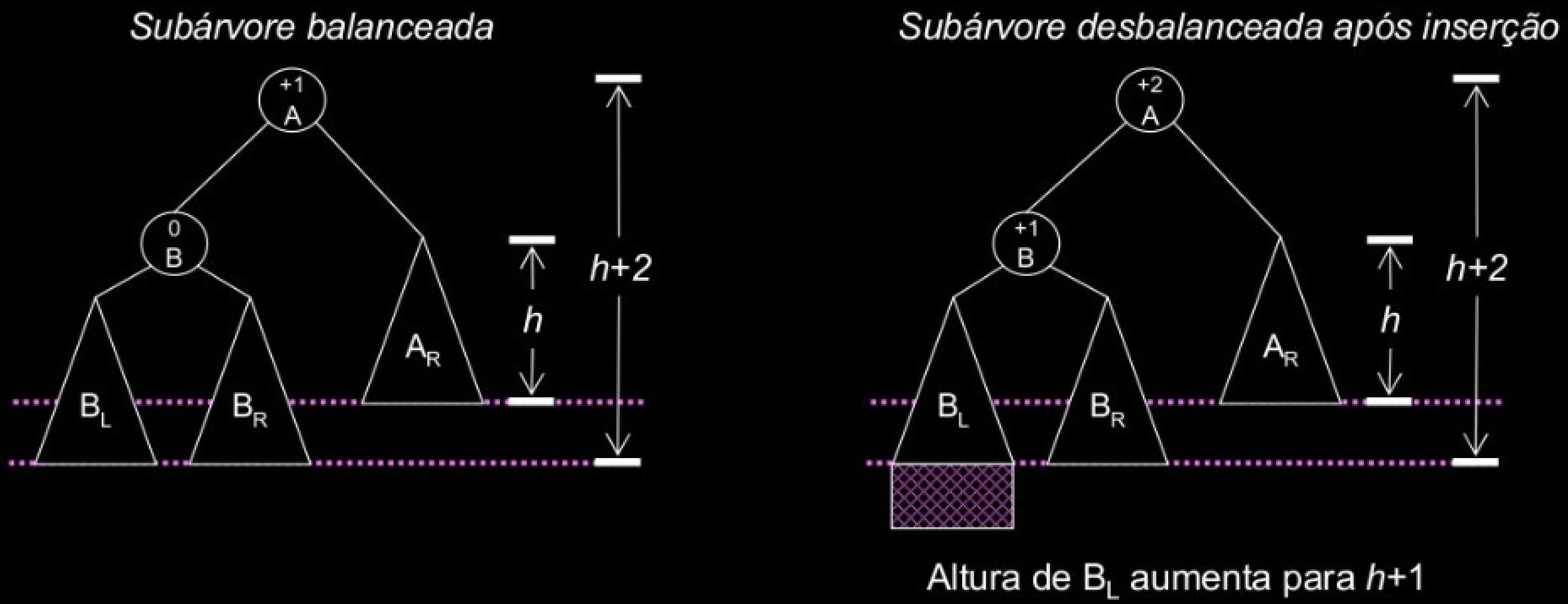
RL: Y inserido na subárvore esquerda da subárvore direita de A

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

Árvore AVL – Rotações Caso LL

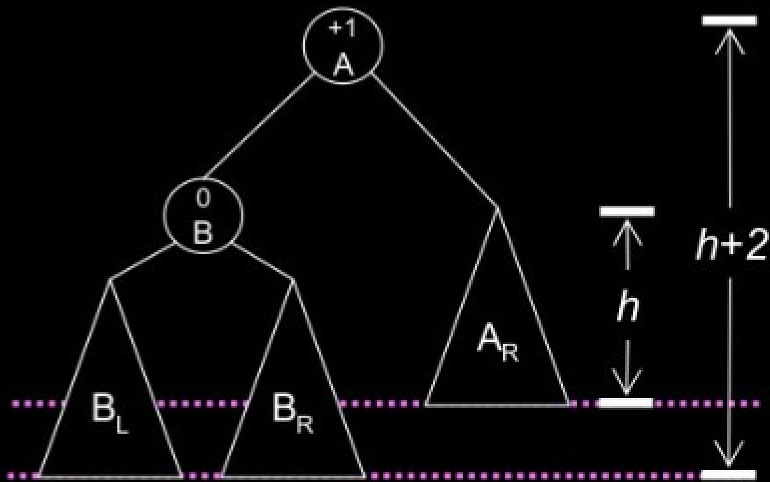


Árvore AVL – Rotações Caso LL

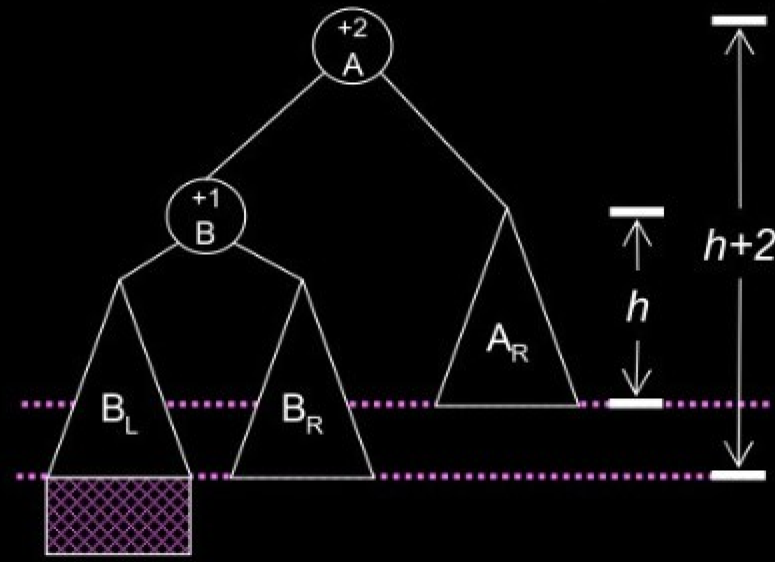


Árvore AVL – Rotações Caso LL

Subárvore balanceada

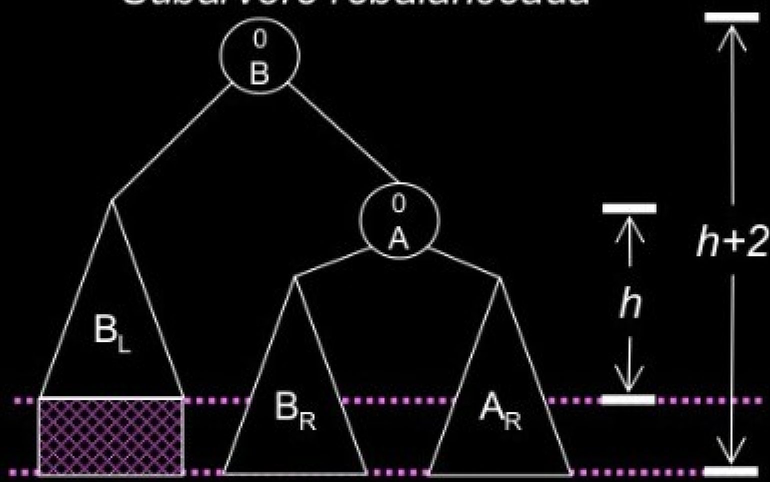


Subárvore desbalanceada após inserção

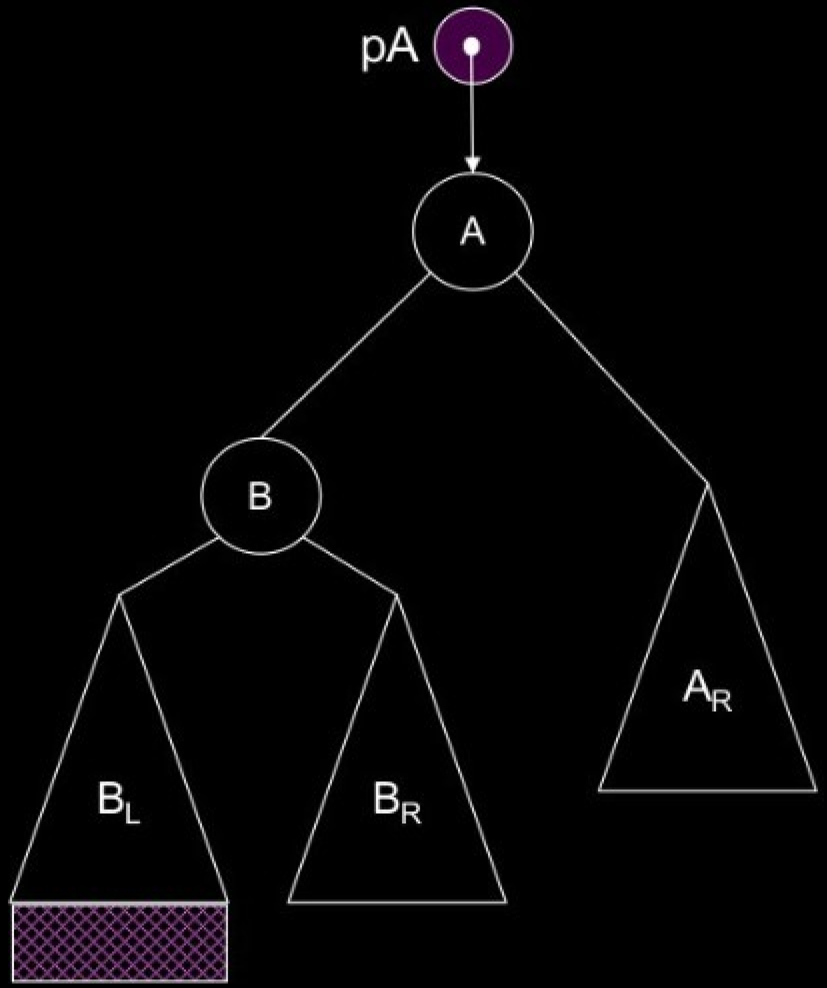


Altura de B_L aumenta para $h+1$

Subárvore rebalanceada

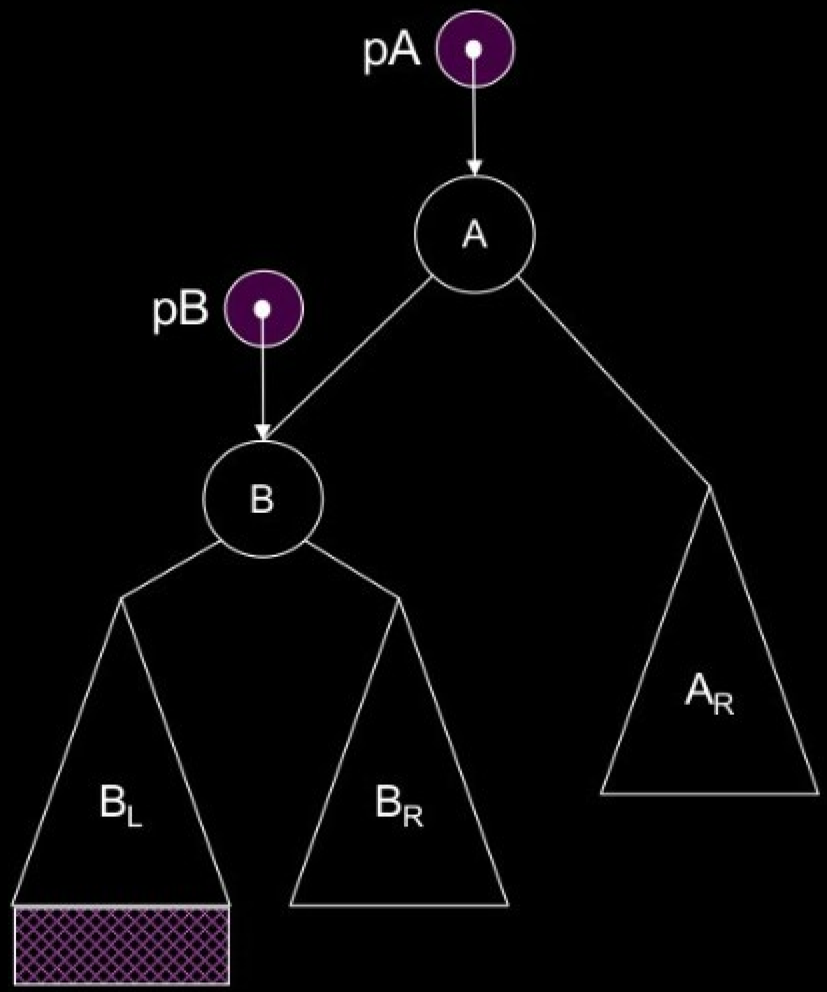


Árvore AVL – Rotações Caso LL



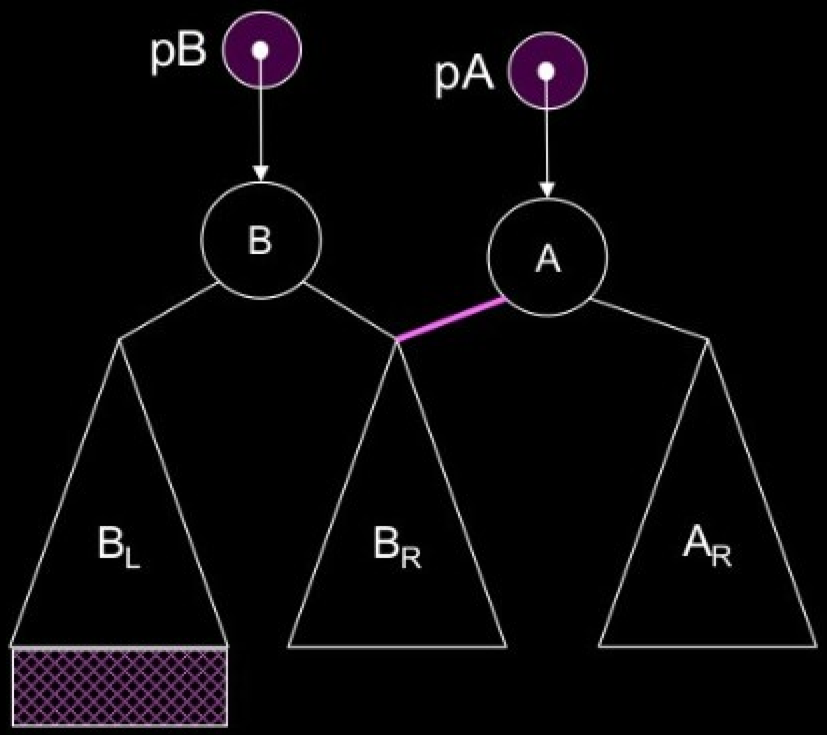
- Assumindo pA e pB ponteiros para as subárvores com raízes A e B:
  - pB = pA->LeftNode;
  - pA->LeftNode = pB->RightNode;
  - pB->RightNode = pA;
  - pA = pB;

Árvore AVL – Rotações Caso LL



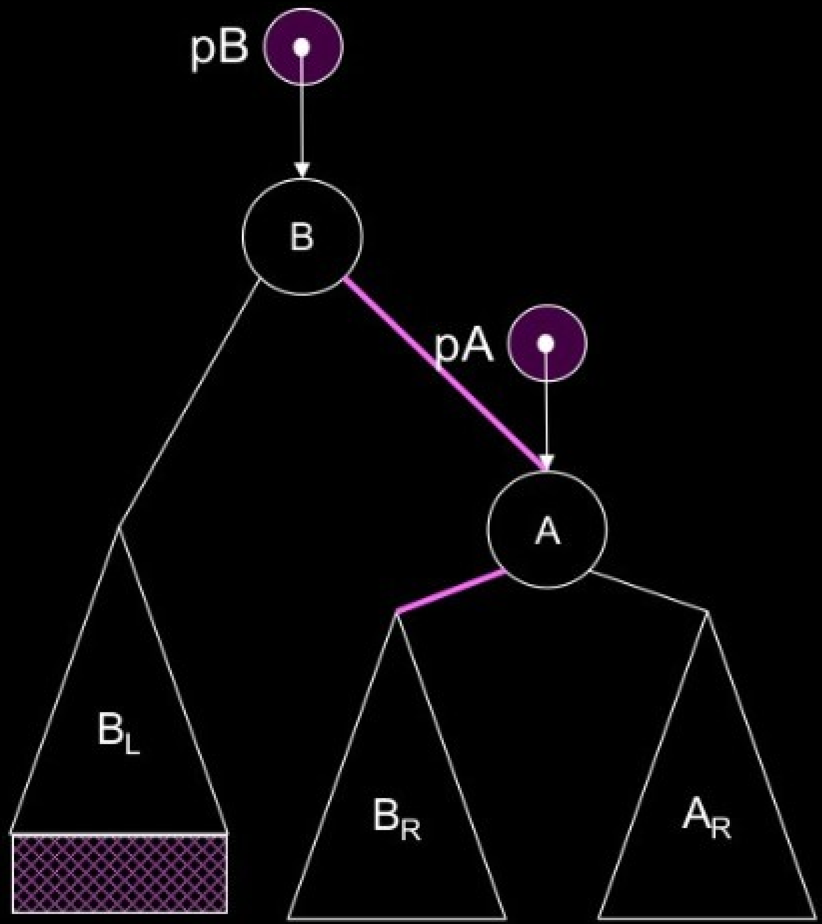
- Assumindo pA e pB ponteiros para as subárvores com raízes A e B:
  - **pB = pA->LeftNode;**
  - pA->LeftNode = pB->RightNode;
  - pB->RightNode = pA;
  - pA = pB;

Árvore AVL – Rotações Caso LL



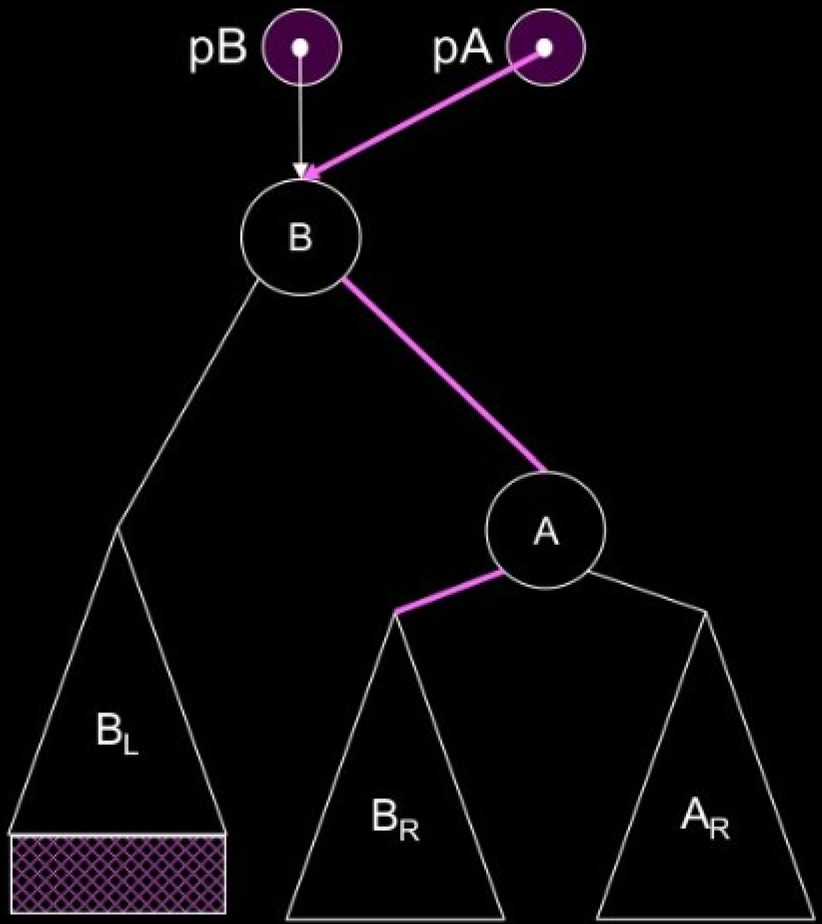
- Assumindo pA e pB ponteiros para as subárvores com raízes A e B:
  - pB = pA->LeftNode;
  - **pA->LeftNode = pB->RightNode;**
  - pB->RightNode = pA;
  - pA = pB;

Árvore AVL – Rotações Caso LL



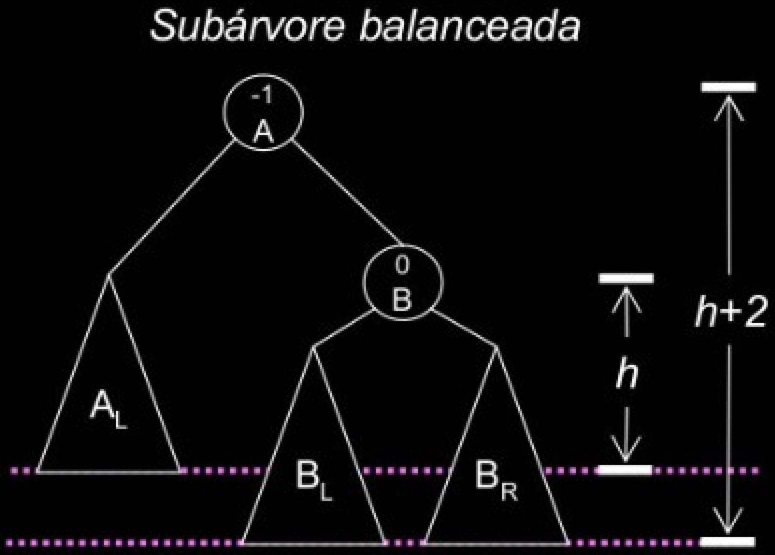
- Assumindo pA e pB ponteiros para as subárvores com raízes A e B:
  - pB = pA->LeftNode;
  - pA->LeftNode = pB->RightNode;
  - **pB->RightNode = pA;**
  - pA = pB;

Árvore AVL – Rotações Caso LL

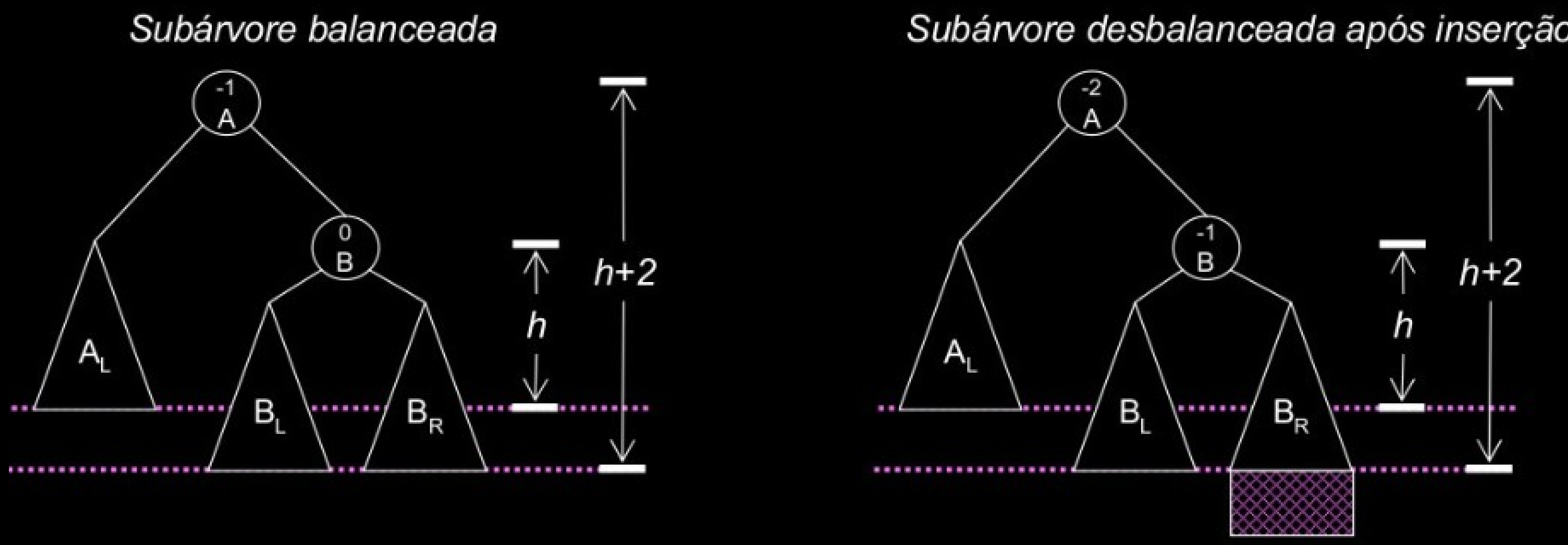


- Assumindo pA e pB ponteiros para as subárvores com raízes A e B:
  - pB = pA->LeftNode;
  - pA->LeftNode = pB->RightNode;
  - pB->RightNode = pA;
  - **pA = pB;**

Árvore AVL – Rotações Caso RR

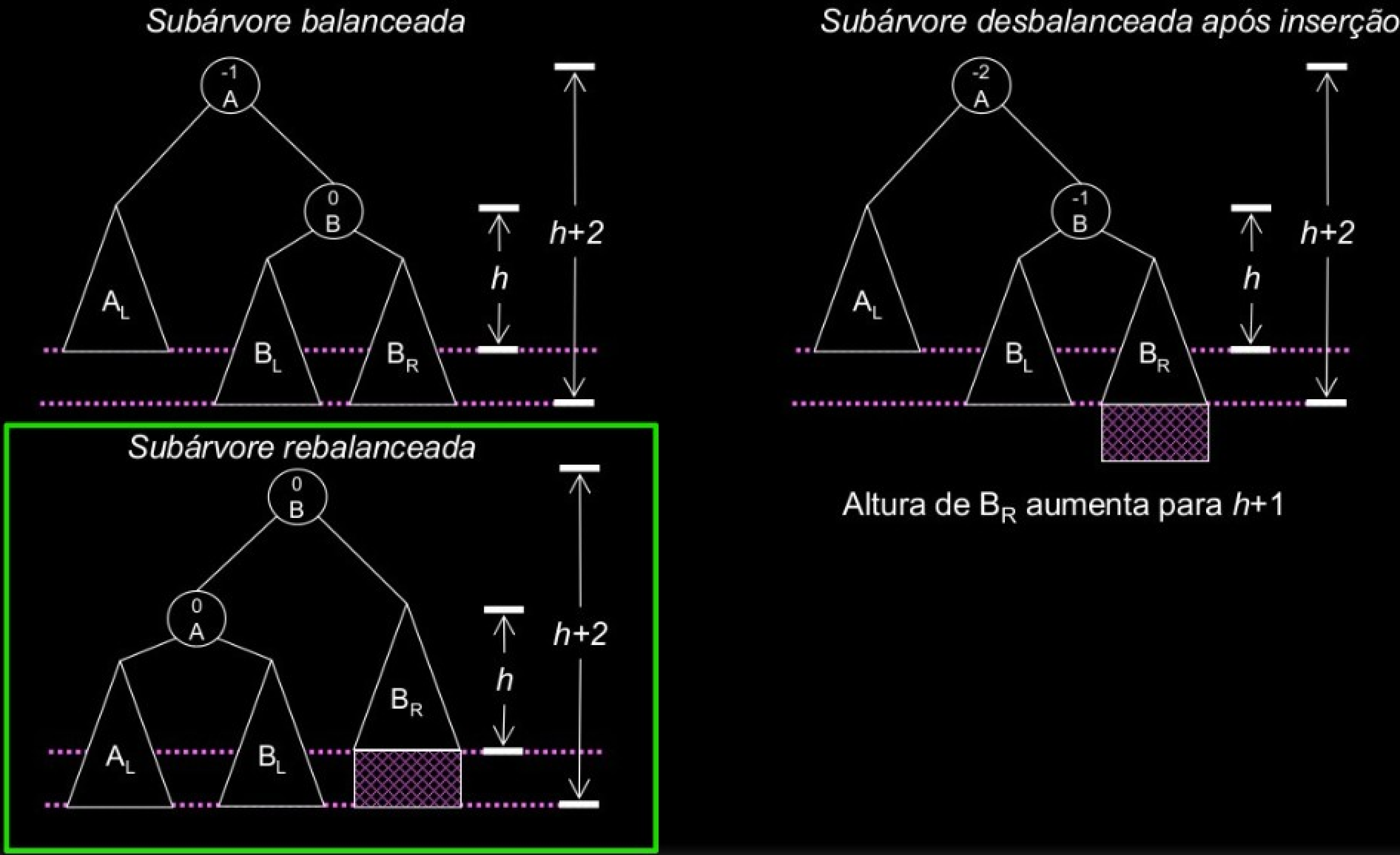


Árvore AVL – Rotações Caso RR

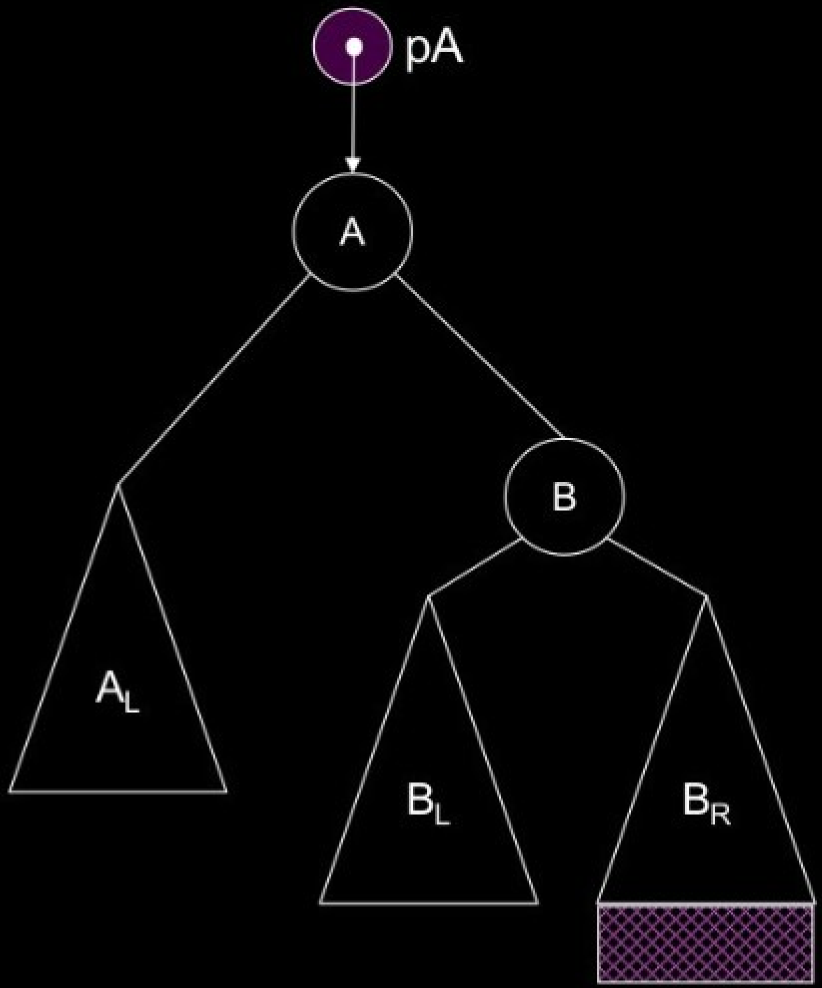


Altura de B_R aumenta para $h+1$

Árvore AVL – Rotações Caso RR

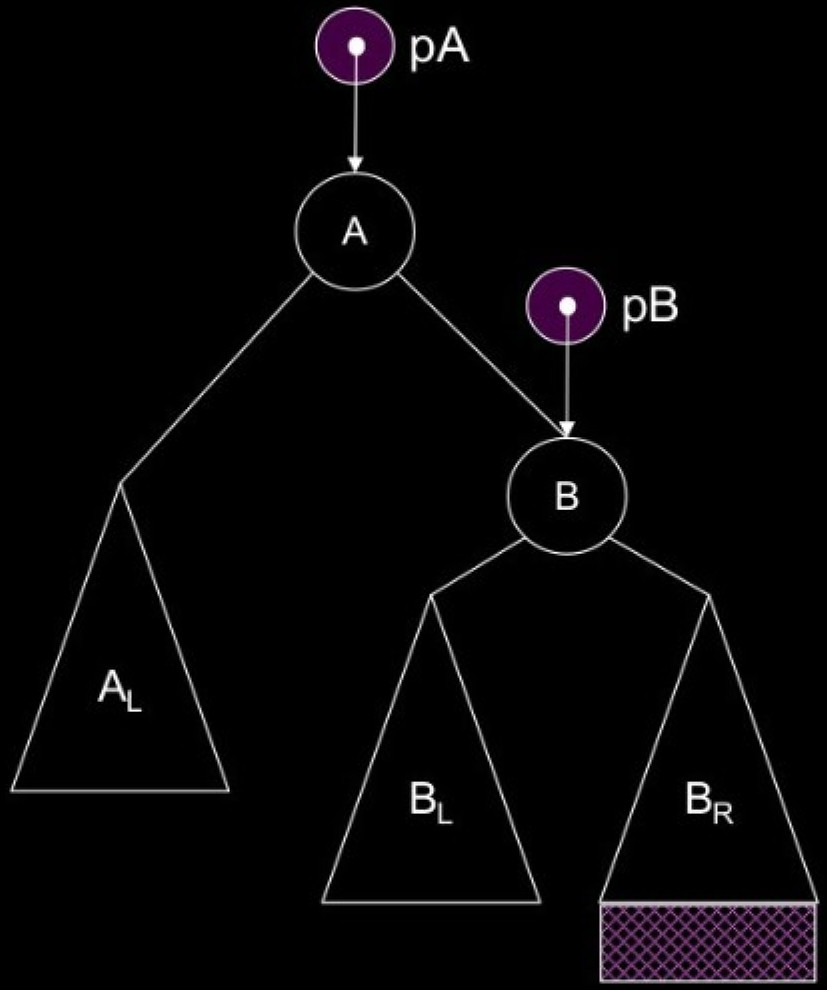


Árvore AVL – Rotações Caso RR



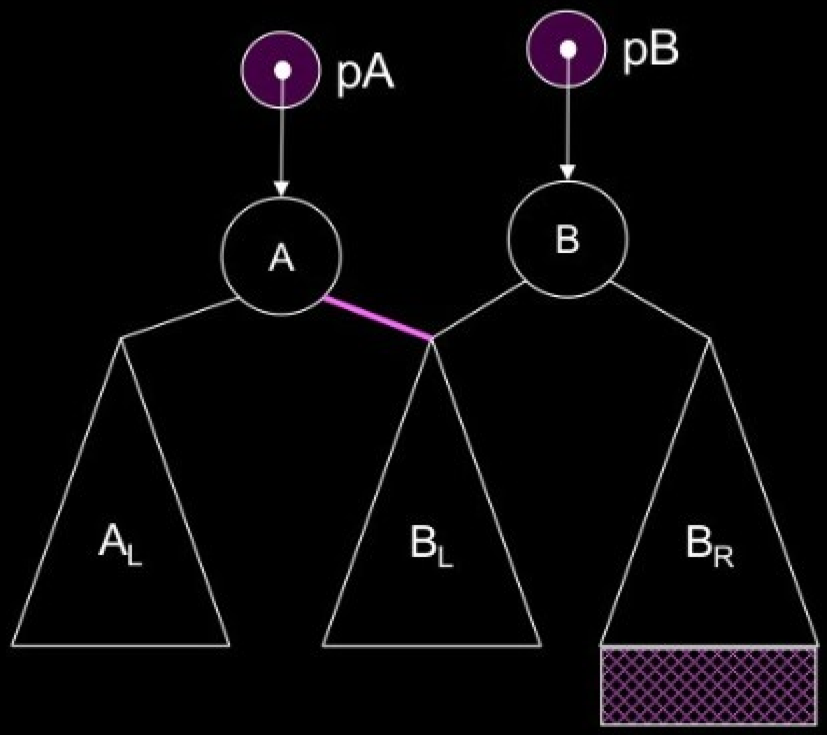
- Assumindo pA e pB ponteiros para as subárvores com raízes A e B:
  - pB = pA->RightNode;
  - pA->RightNode = pB->LeftNode;
  - pB->LeftNode = pA;
  - pA = pB;

Árvore AVL – Rotações Caso RR



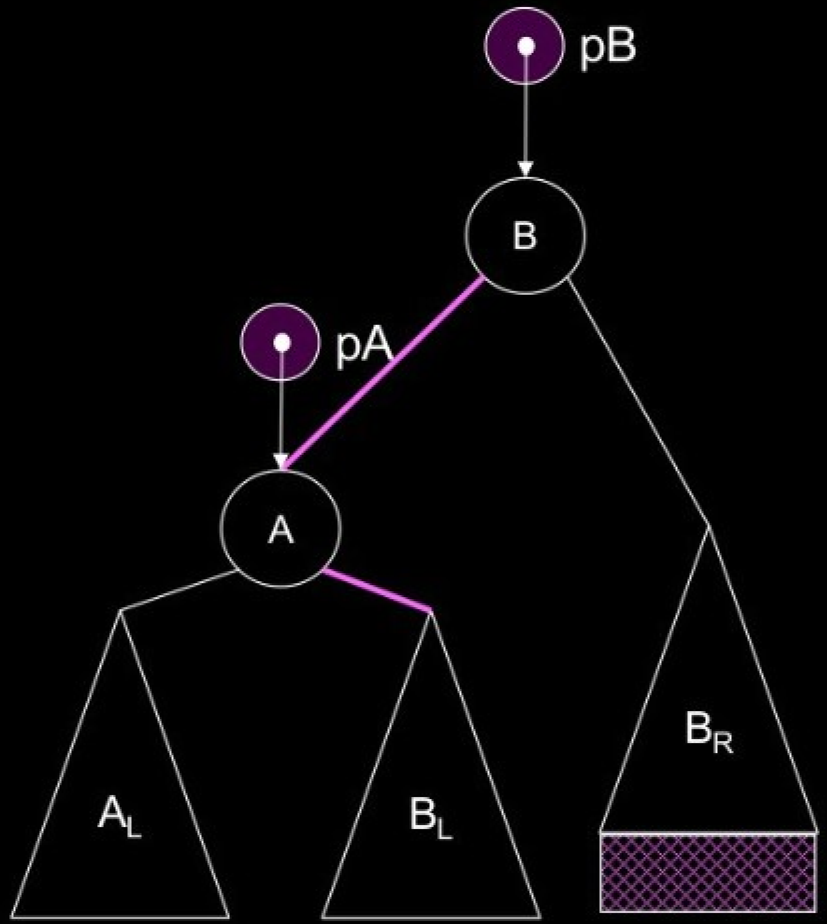
- Assumindo pA e pB ponteiros para as subárvores com raízes A e B:
  - **pB = pA->RightNode;**
  - pA->RightNode = pB->LeftNode;
  - pB->LeftNode = pA;
  - pA = pB;

Árvore AVL – Rotações Caso RR



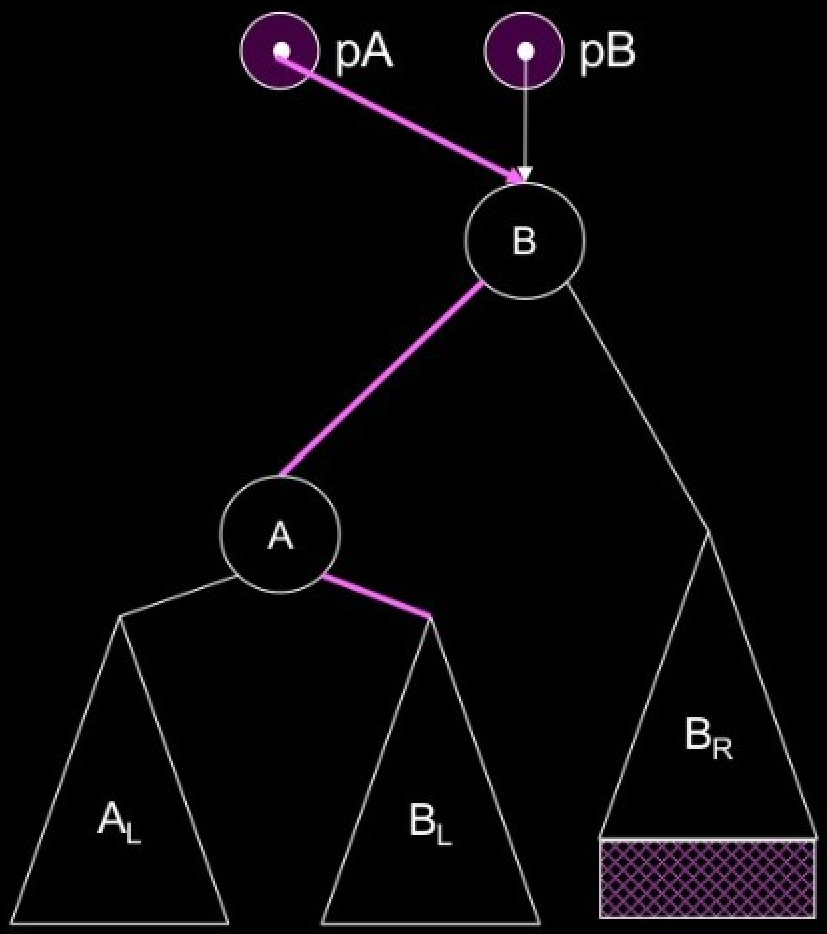
- Assumindo pA e pB ponteiros para as subárvores com raízes A e B:
  - pB = pA->RightNode;
  - **pA->RightNode = pB->LeftNode;**
  - pB->LeftNode = pA;
  - pA = pB;

Árvore AVL – Rotações Caso 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;

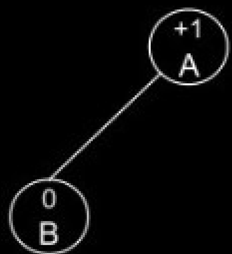
Árvore AVL – Rotações Caso RR



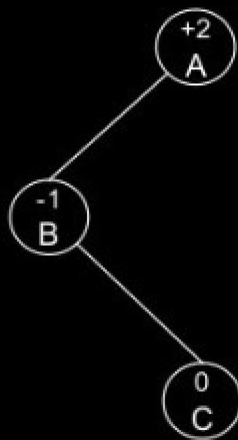
- Assumindo pA e pB ponteiros para as subárvores com raízes A e B:
  - pB = pA->RightNode;
  - pA->RightNode = pB->LeftNode;
  - pB->LeftNode = pA;
  - **pA = pB;**

Árvore AVL – Rotações Caso LR a)

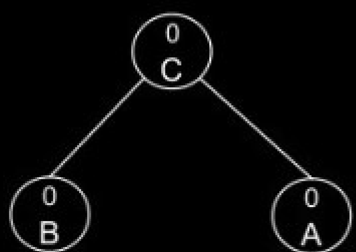
Subárvore balanceada



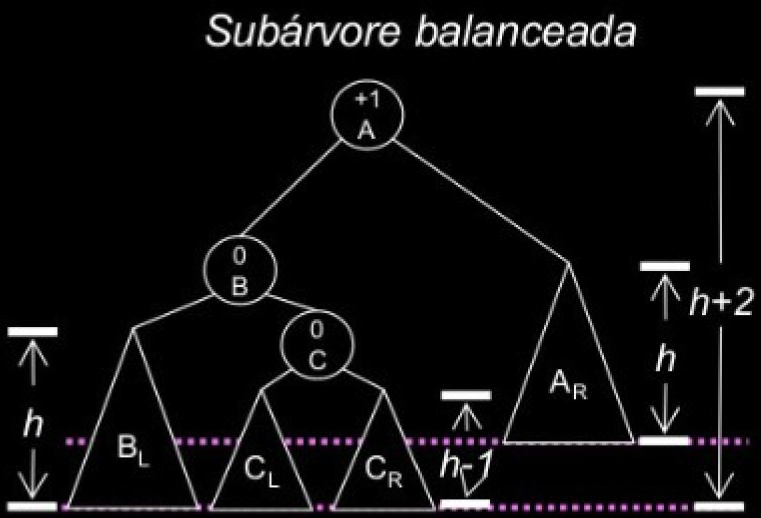
Subárvore desbalanceada após inserção



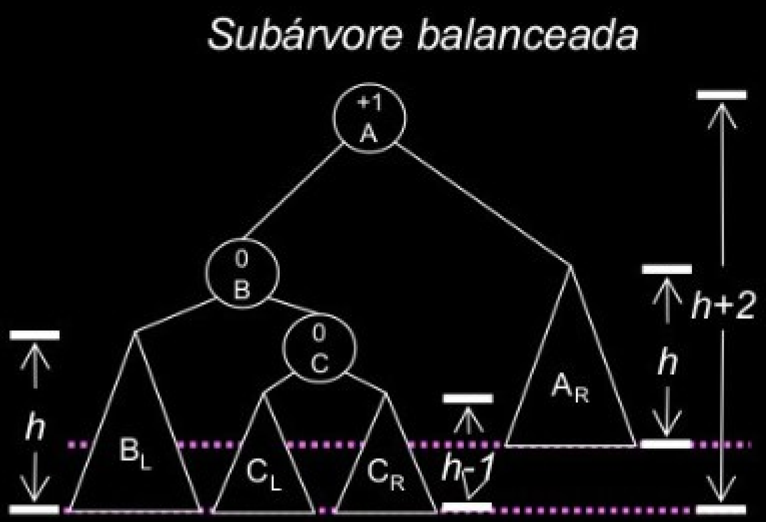
Subárvore rebalanceada

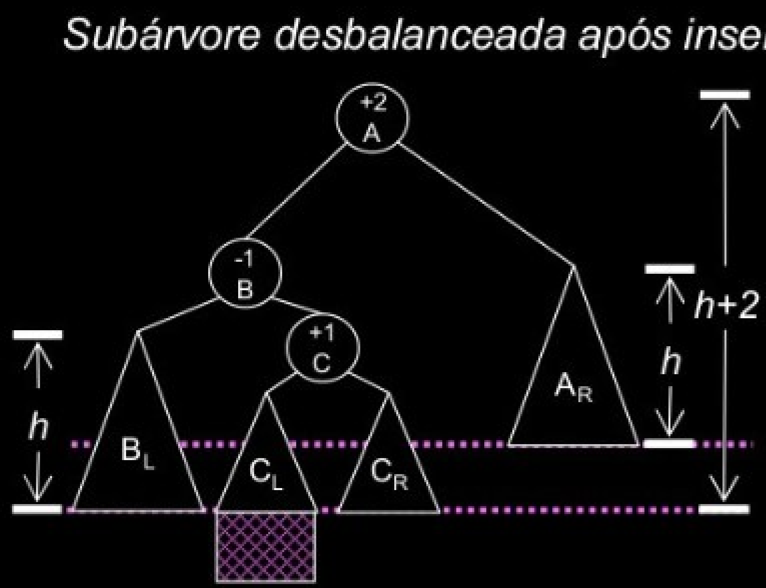


Árvore AVL – Rotações Caso LR b)

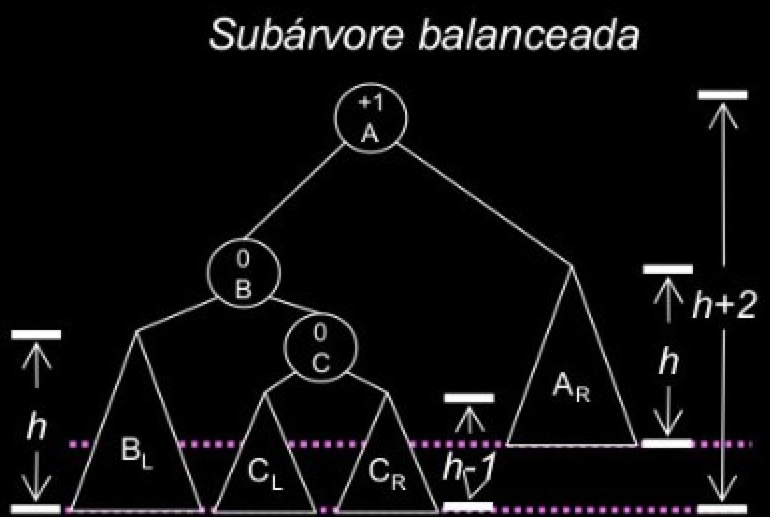


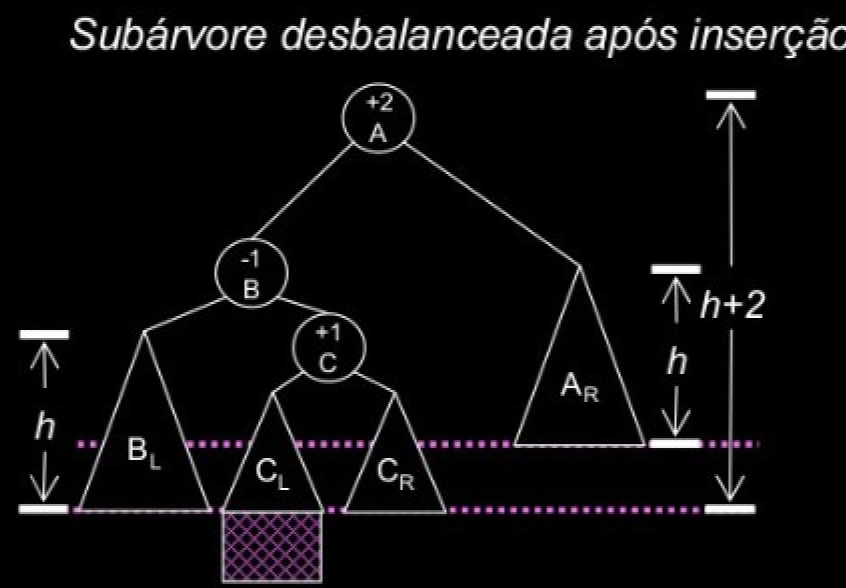
Árvore AVL – Rotações Caso LR b)

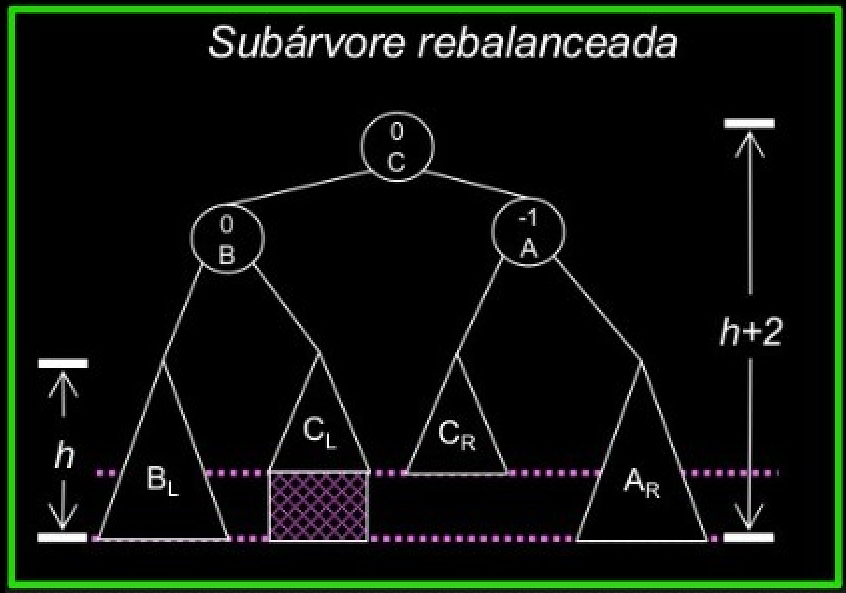




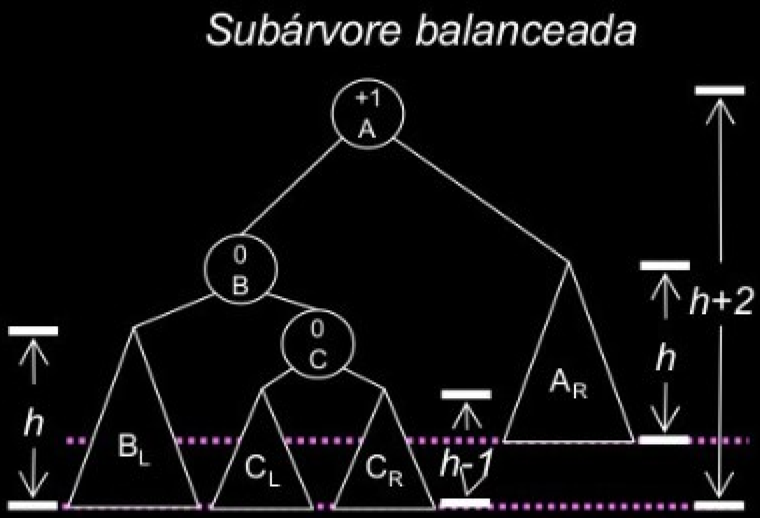
Árvore AVL – Rotações Caso LR b)



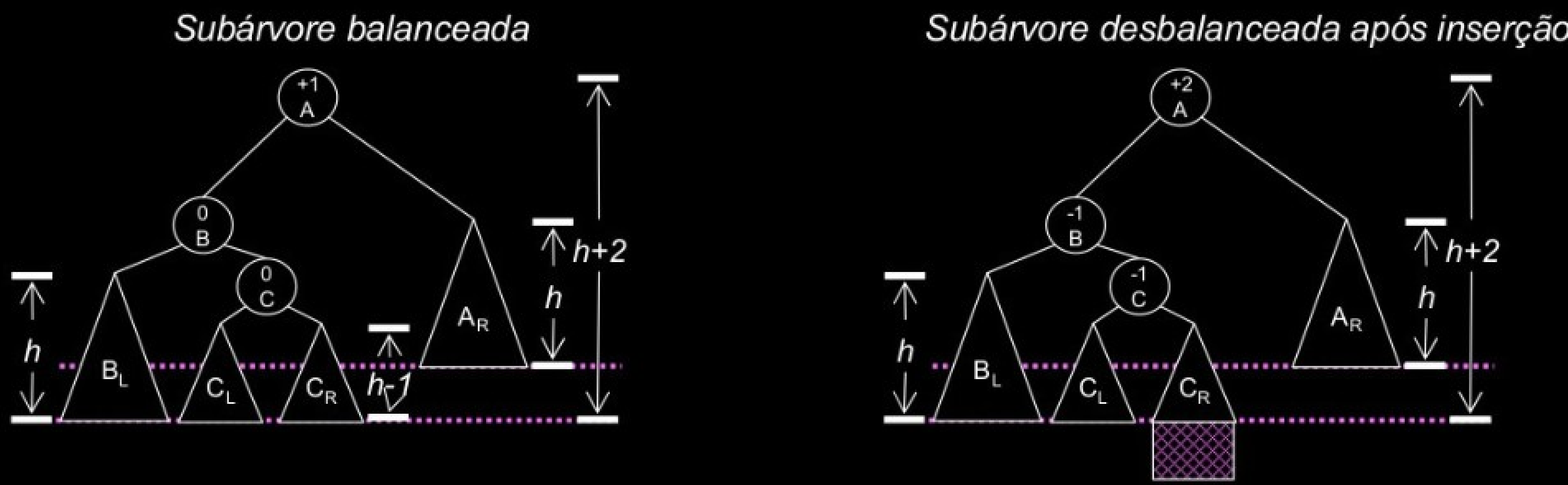




Árvore AVL – Rotações Caso LR c)

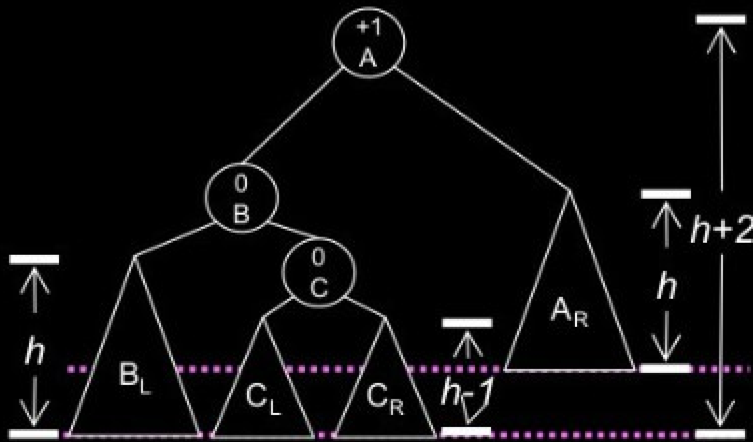


Árvore AVL – Rotações Caso LR c)

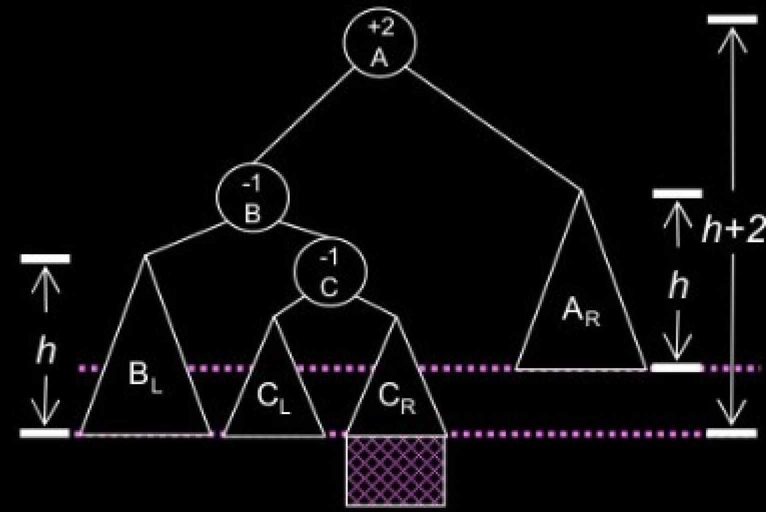


Árvore AVL – Rotações Caso LR c)

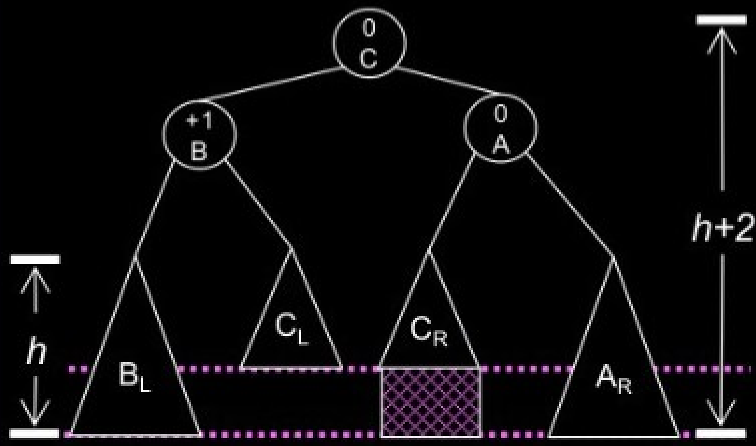
Subárvore balanceada



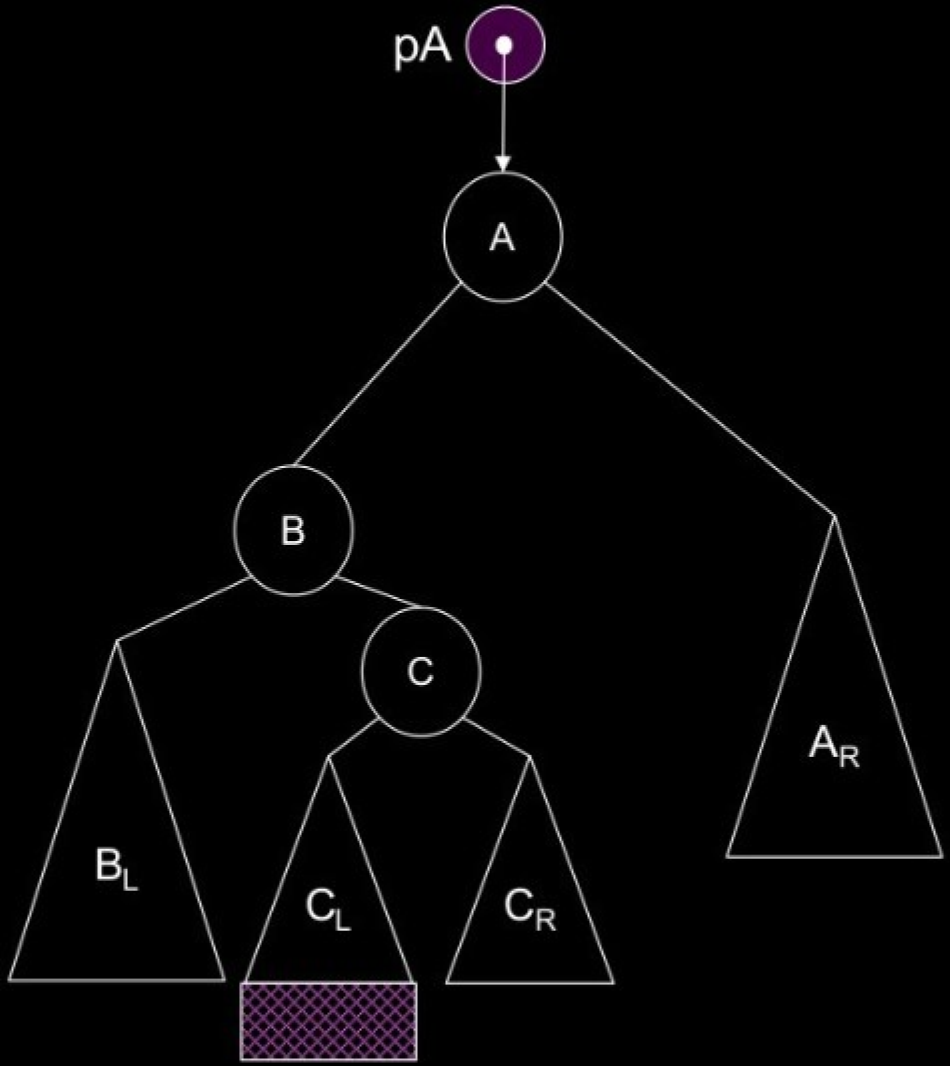
Subárvore desbalanceada após inserção



Subárvore rebalanceada

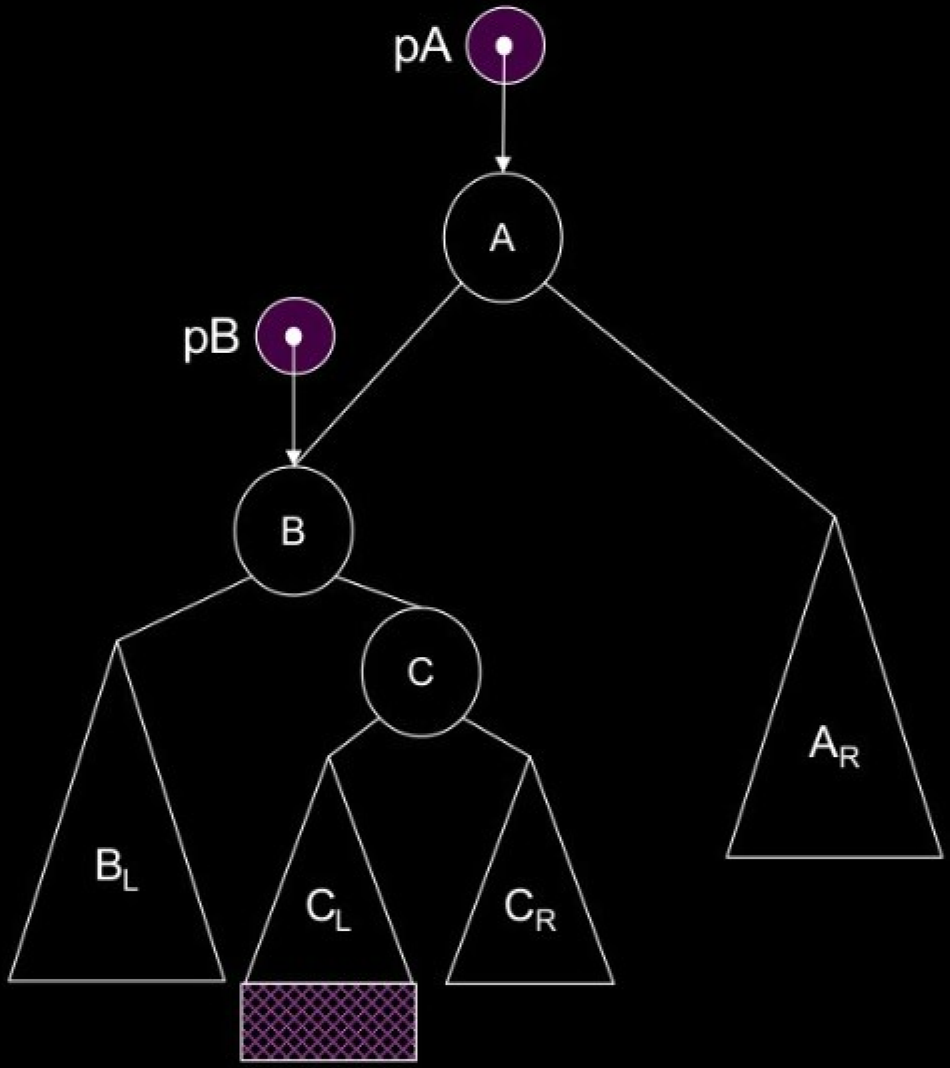


Árvore AVL – Rotações Caso LR



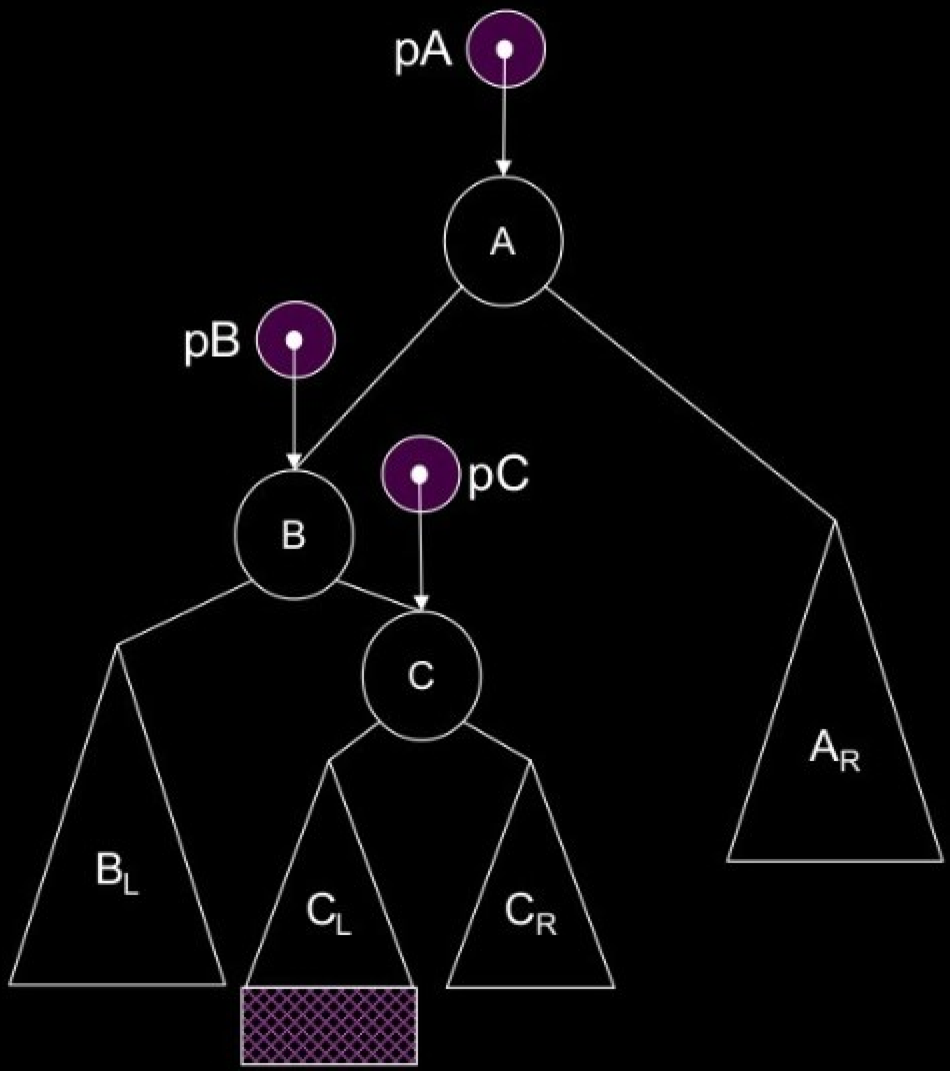
- Assumindo pA, pB e pC ponteiros para as subárvores com raízes A, B e C:
  - pB = pA->LeftNode;
  - pC = pB->RightNode;
  - pB->RightNode = pC->LeftNode;
  - pC->LeftNode = pB;
  - pA->LeftNode = pC->RightNode;
  - pC->RightNode = pA;
  - pA = pC;

Árvore AVL – Rotações Caso LR



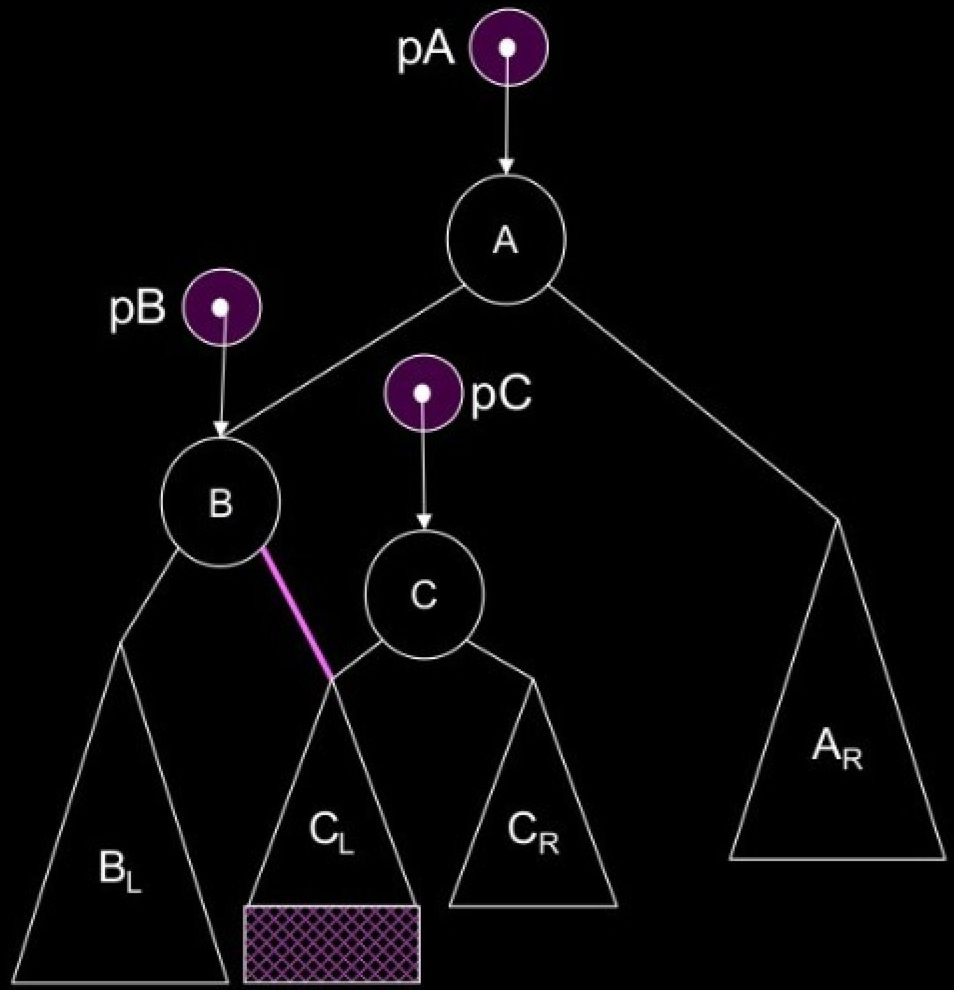
- Assumindo pA, pB e pC ponteiros para as subárvores com raízes A, B e C:
  - **pB = pA->LeftNode;**
  - pC = pB->RightNode;
  - pB->RightNode = pC->LeftNode;
  - pC->LeftNode = pB;
  - pA->LeftNode = pC->RightNode;
  - pC->RightNode = pA;
  - pA = pC;

Árvore AVL – Rotações Caso LR



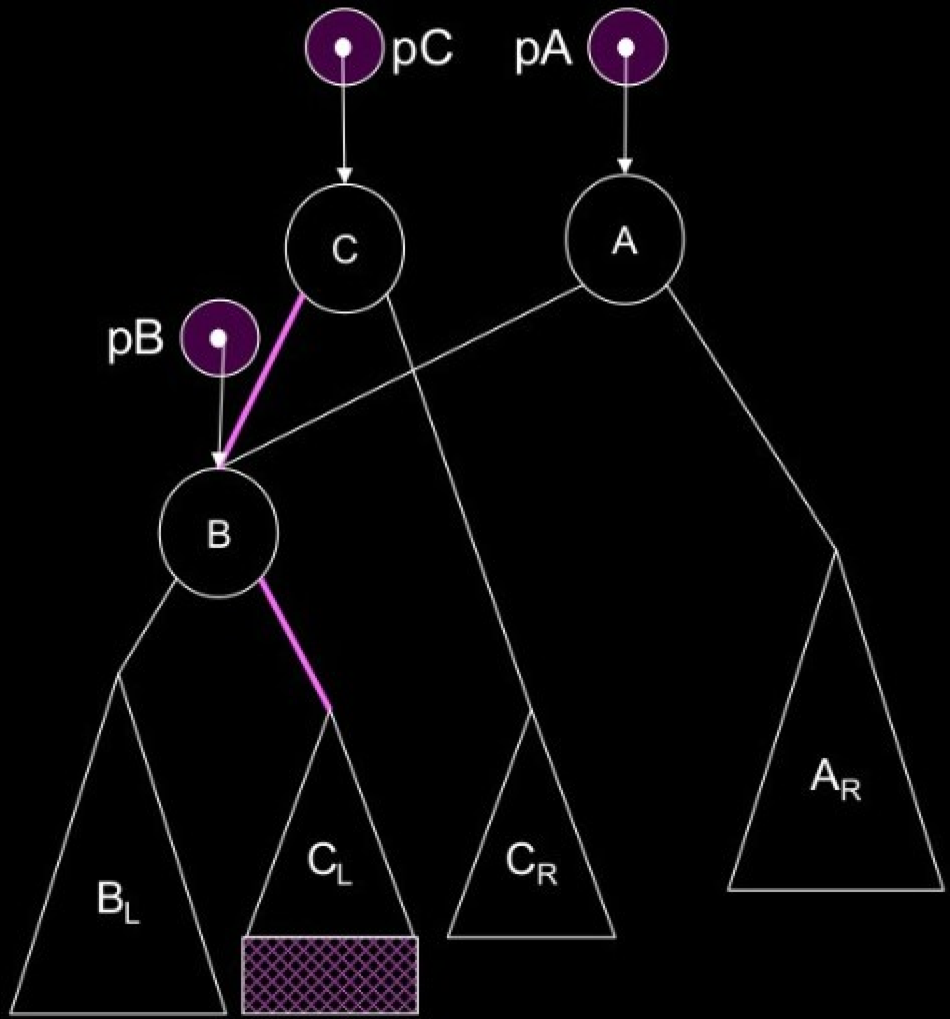
- Assumindo pA, pB e pC ponteiros para as subárvores com raízes A, B e C:
  - pB = pA->LeftNode;
  - **pC = pB->RightNode;**
  - pB->RightNode = pC->LeftNode;
  - pC->LeftNode = pB;
  - pA->LeftNode = pC->RightNode;
  - pC->RightNode = pA;
  - pA = pC;

Árvore AVL – Rotações Caso LR



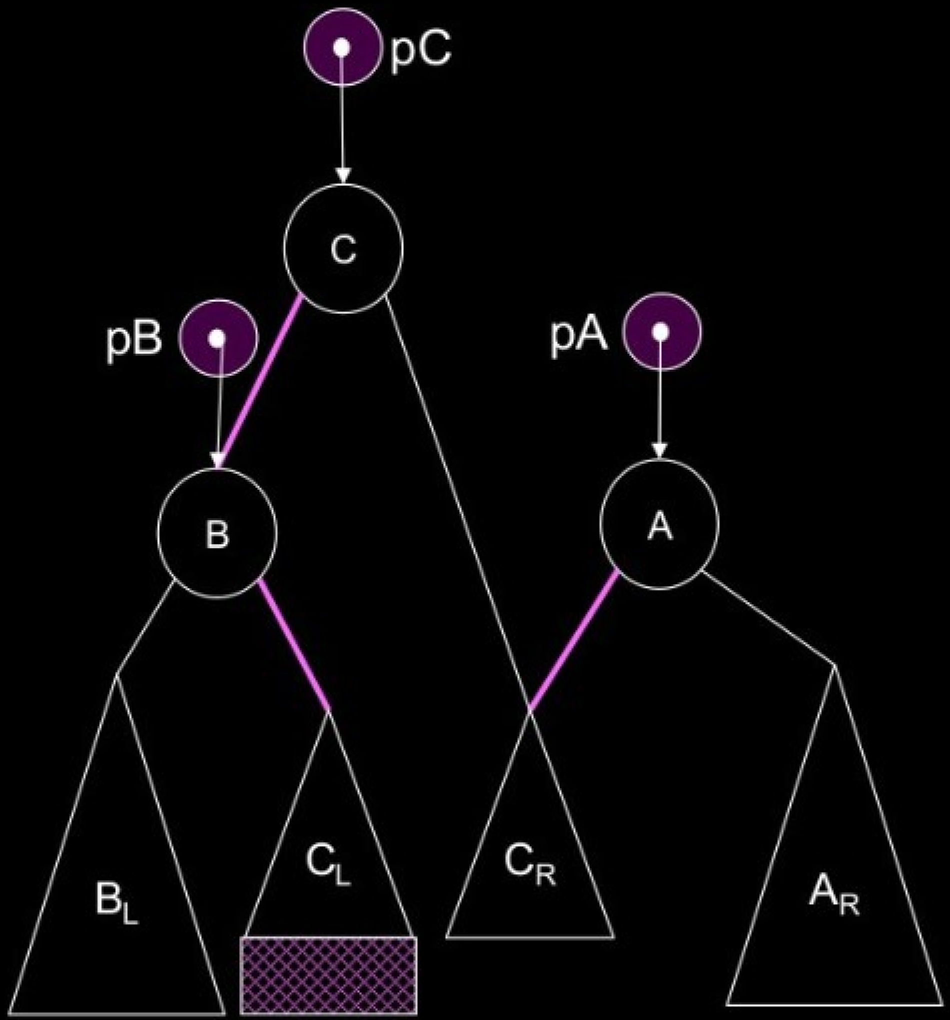
- Assumindo pA, pB e pC ponteiros para as subárvores com raízes A, B e C:
  - pB = pA->LeftNode;
  - pC = pB->RightNode;
  - **pB->RightNode = pC->LeftNode;**
  - pC->LeftNode = pB;
  - pA->LeftNode = pC->RightNode;
  - pC->RightNode = pA;
  - pA = pC;

Árvore AVL – Rotações Caso LR



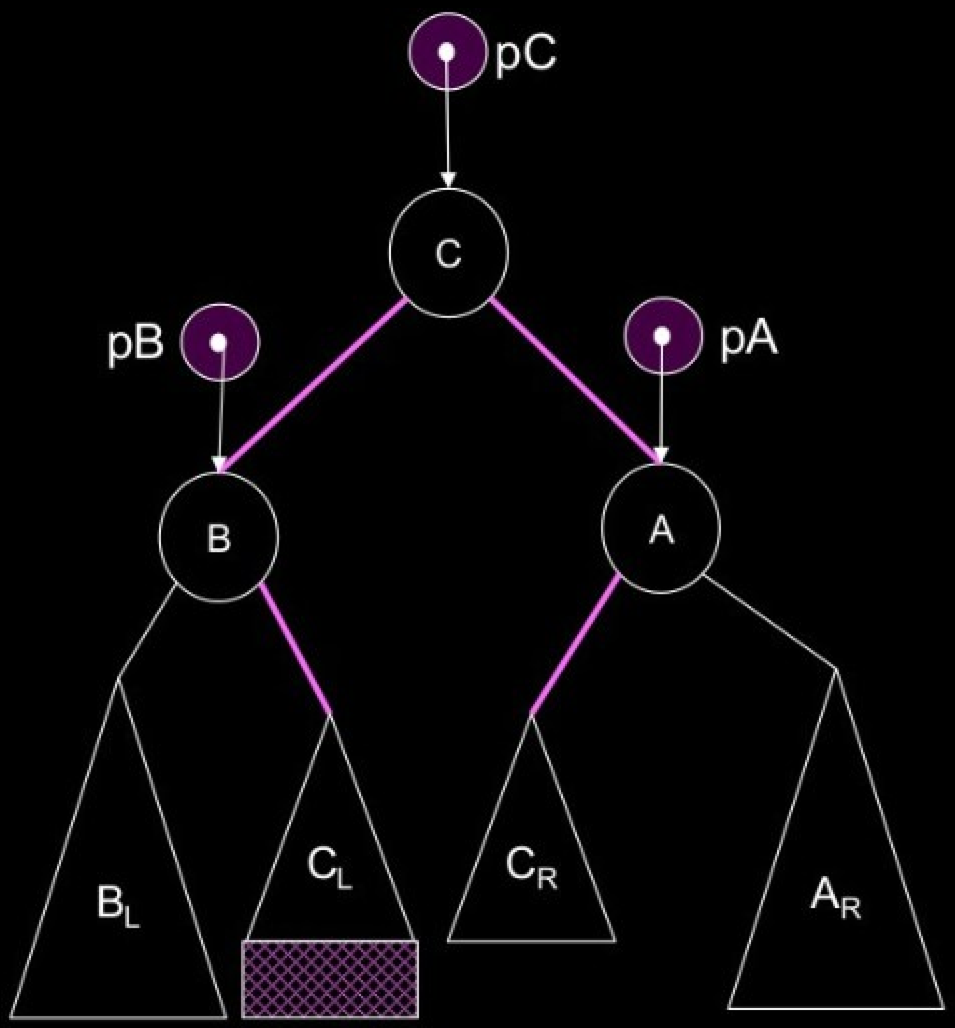
- Assumindo pA, pB e pC ponteiros para as subárvores com raízes A, B e C:
  - pB = pA->LeftNode;
  - pC = pB->RightNode;
  - pB->RightNode = pC->LeftNode;
  - **pC->LeftNode = pB;**
  - pA->LeftNode = pC->RightNode;
  - pC->RightNode = pA;
  - pA = pC;

Árvore AVL – Rotações Caso LR



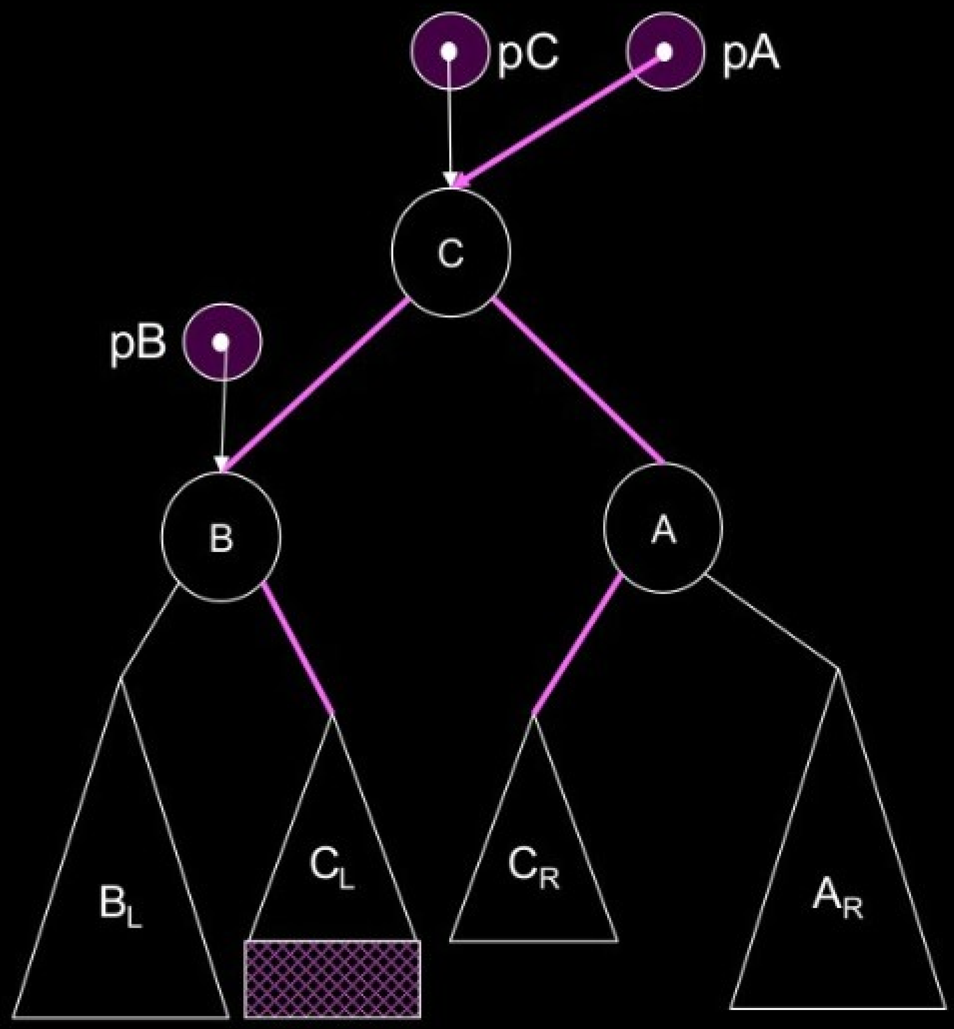
- Assumindo pA, pB e pC ponteiros para as subárvores com raízes A, B e C:
  - pB = pA->LeftNode;
  - pC = pB->RightNode;
  - pB->RightNode = pC->LeftNode;
  - pC->LeftNode = pB;
  - **pA->LeftNode = pC->RightNode;**
  - pC->RightNode = pA;
  - pA = pC;

Árvore AVL – Rotações Caso 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;

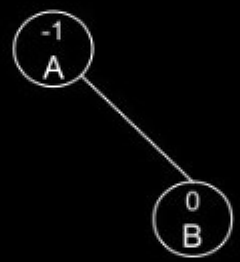
Árvore AVL – Rotações Caso LR



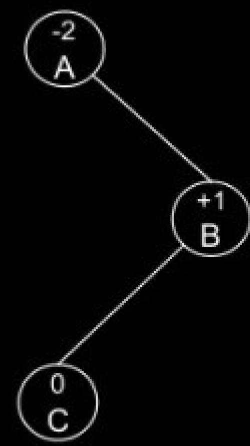
- Assumindo pA, pB e pC ponteiros para as subárvores com raízes A, B e C:
  - pB = pA->LeftNode;
  - pC = pB->RightNode;
  - pB->RightNode = pC->LeftNode;
  - pC->LeftNode = pB;
  - pA->LeftNode = pC->RightNode;
  - pC->RightNode = pA;
  - **pA = pC;**

Árvore AVL – Rotações Caso RL a)

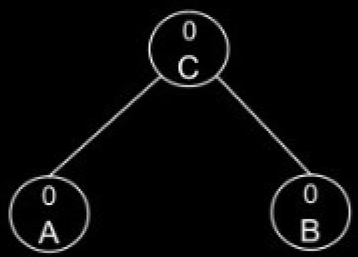
Subárvore balanceada



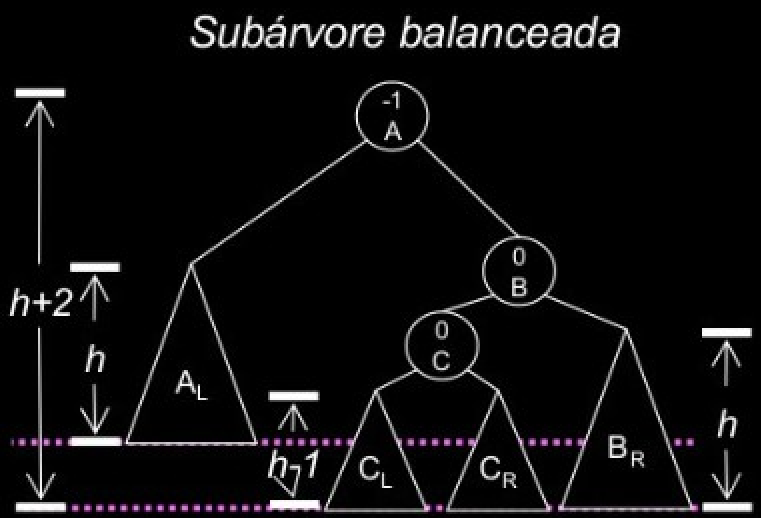
Subárvore desbalanceada após inserção

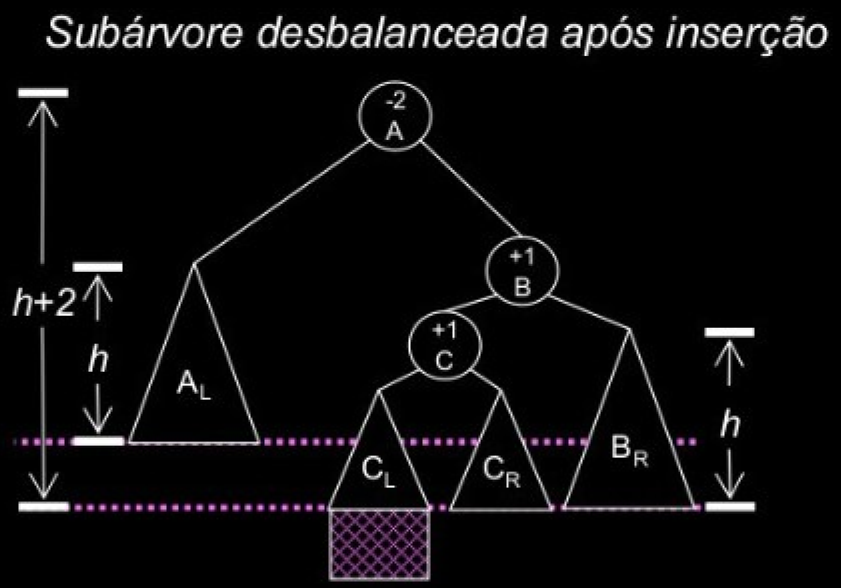


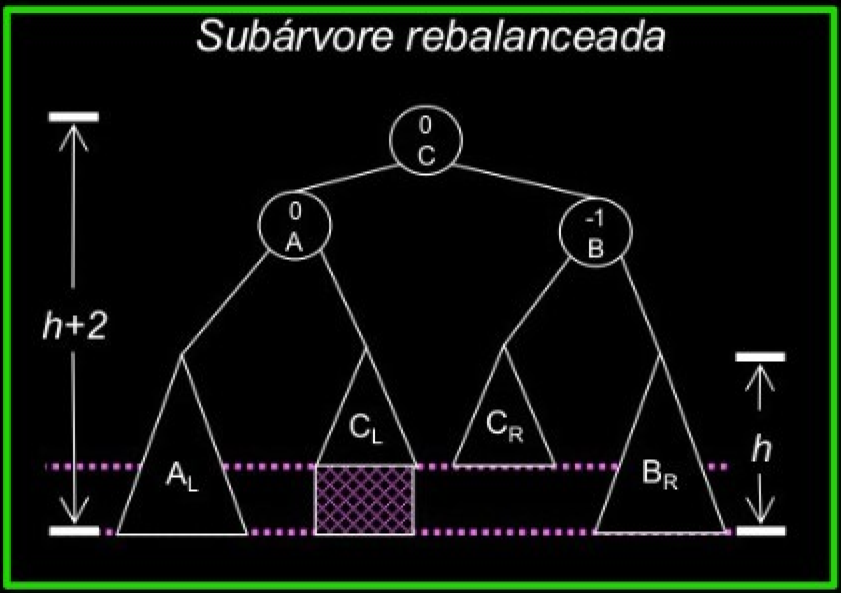
Subárvore rebalanceada



Árvore AVL – Rotações Caso RL b)

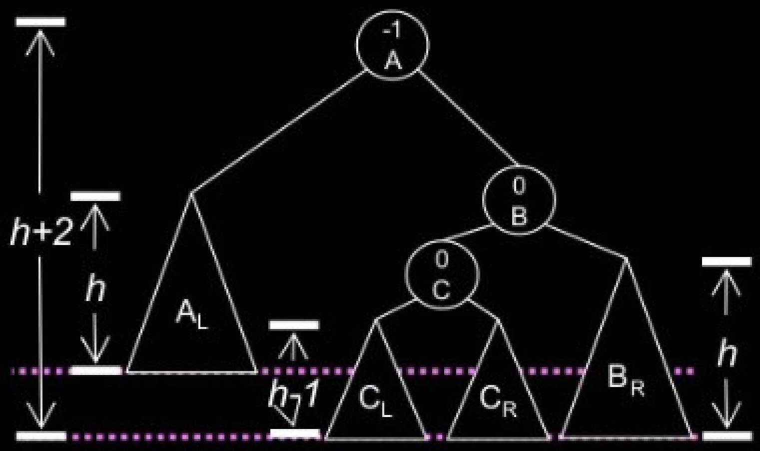




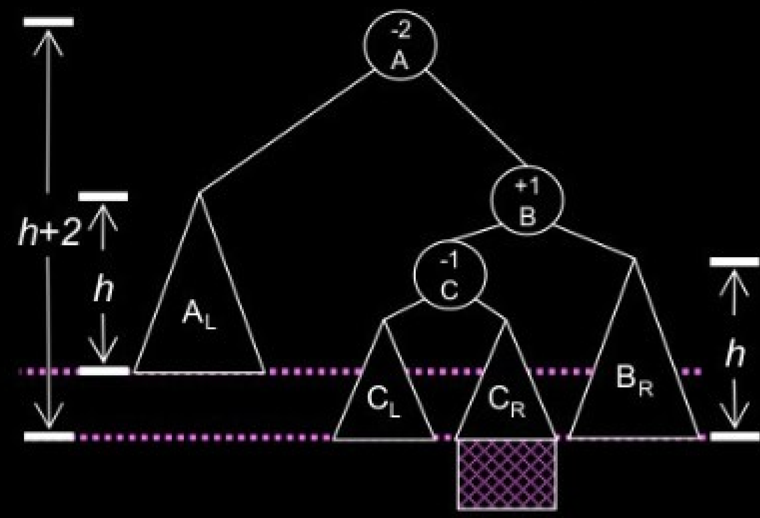


Árvore AVL – Rotações Caso RL c)

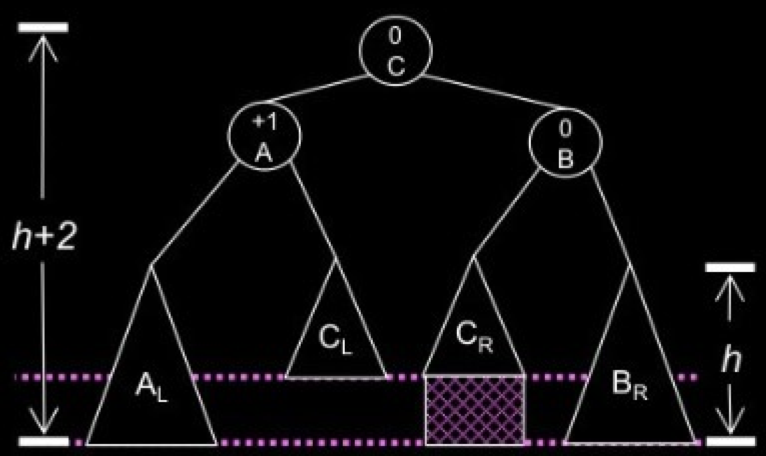
Subárvore balanceada



Subárvore desbalanceada após inserção



Subárvore rebalanceada

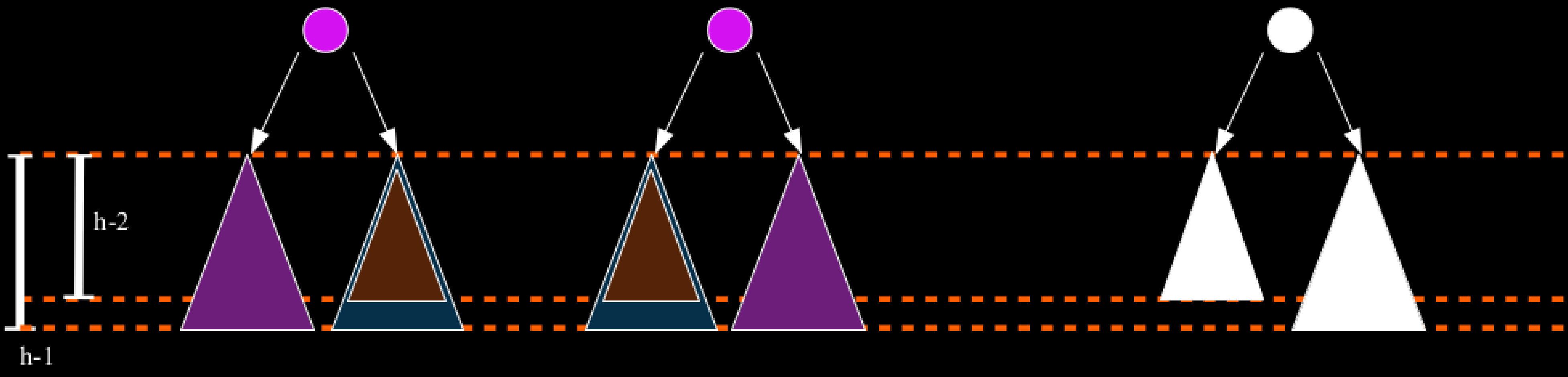


Balanceamento de árvores AVL

- Uma árvore AVL de altura h é balanceada se h = O(log(n))
- Outra forma de pensar: Dada uma árvore AVL de altura h, qual seria o valor mínimo possivel para n?

Balanceamento de árvores AVL

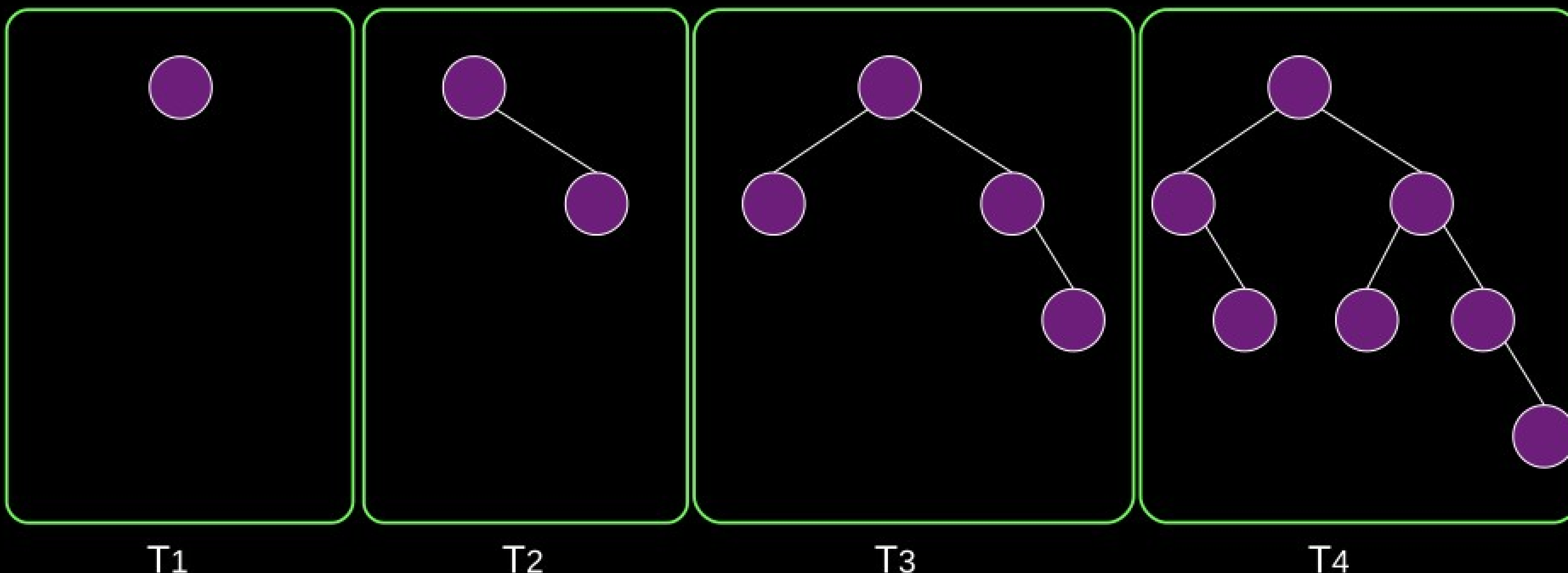
- Uma árvore AVL de altura h é balanceada se h = O(log(n))
- Outra forma de pensar: Dada uma árvore AVL de altura h, qual seria o valor mínimo possivel para n?



Balanceamento de árvores AVL

- Seja T_h uma árvore AVL com altura h e número mínimo de nós.

Nesta definição → h=4



Balanceamento de árvores AVL

- Basta calcular um limite inferior do número de nós de T_h . Seja $|T_h|$ o número de nós de T_h.

$$\begin{cases} |T_h| = 0 & , \text{ para } h = 0 \\ |T_h| = 1 & , \text{ para } h = 1 \\ |T_h| = 1 + |T_{h-1}| + |T_{h-2}| & , \text{ para } h > 1 \end{cases}$$

h	$\|T_h\|$
0	0
1	1
2	2
3	4
4	7
5	12
6	20
7	33
8	54
9	88
10	143

Sequencia de Fibonacci:

$$\begin{cases} f_h = 0 & \text{, para } h = 0 \\ f_h = 1 & \text{, para } h = 1 \\ f_h = f_{h-1} + f_{h-2} & \text{, para } h > 1 \end{cases}$$

$$f_h = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^h - \left(\frac{1-\sqrt{5}}{2}\right)^h\right]$$

$$|T_h| \geq f_h$$

Balanceamento de árvores AVL

$|T_h| \geq f_h$

$$f_h = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^h - \left(\frac{1-\sqrt{5}}{2}\right)^h\right]$$

Como $\frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^h < 1$ Temos $|T_h| > \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^h - 1$

Balanceamento de árvores AVL

$|T_h| \geq f_h$

$$f_h = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^h - \left(\frac{1-\sqrt{5}}{2}\right)^h\right]$$

Como $\frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^h < 1$ Temos $|T_h| > \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^h - 1$

$$h < \frac{1}{\log_2 a}\log_2(|T_h|+1) + \log_a(\sqrt{5}) \qquad a = \frac{1+\sqrt{5}}{2}$$

$$h = O(\log n)$$

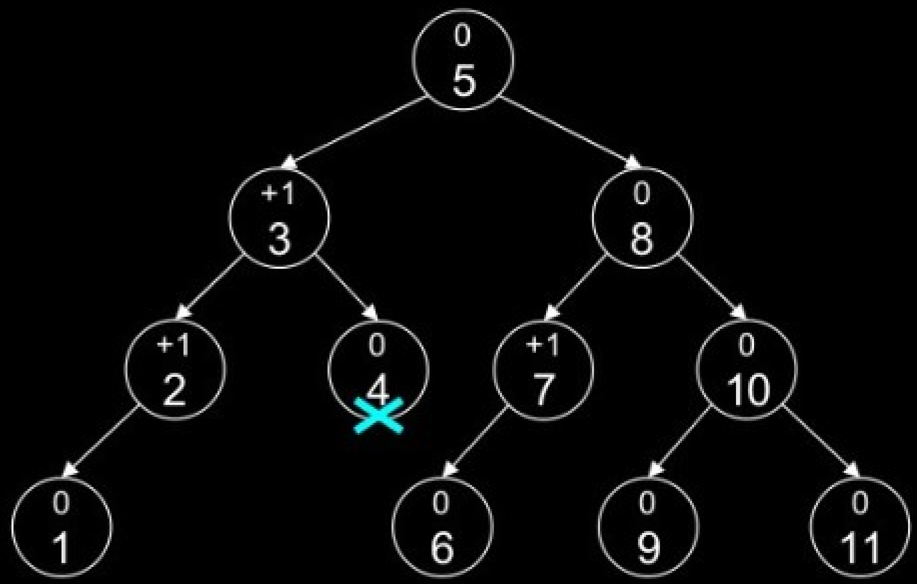
AVL é uma árvore balanceada!

Remoção em árvores AVL

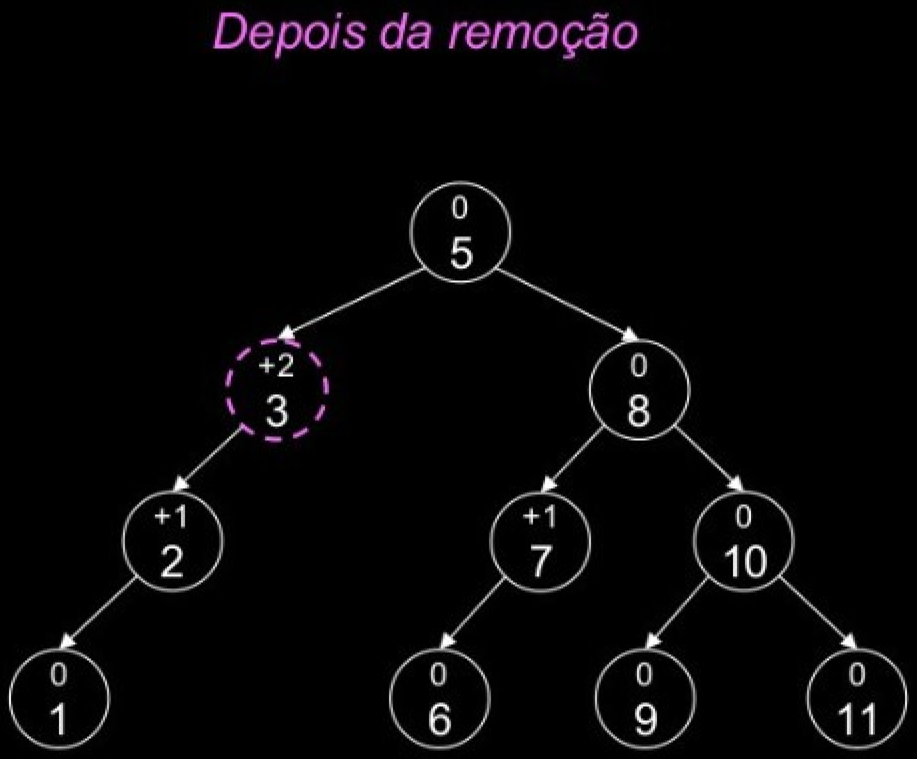
- A remoção em árvores AVL é similar à remoção em uma Árvore Binária de Busca (ABB).
- Todavia, é preciso verificar o balanceamento e, se necessário, aplicar algumas das rotações.

Remoção em árvores AVL

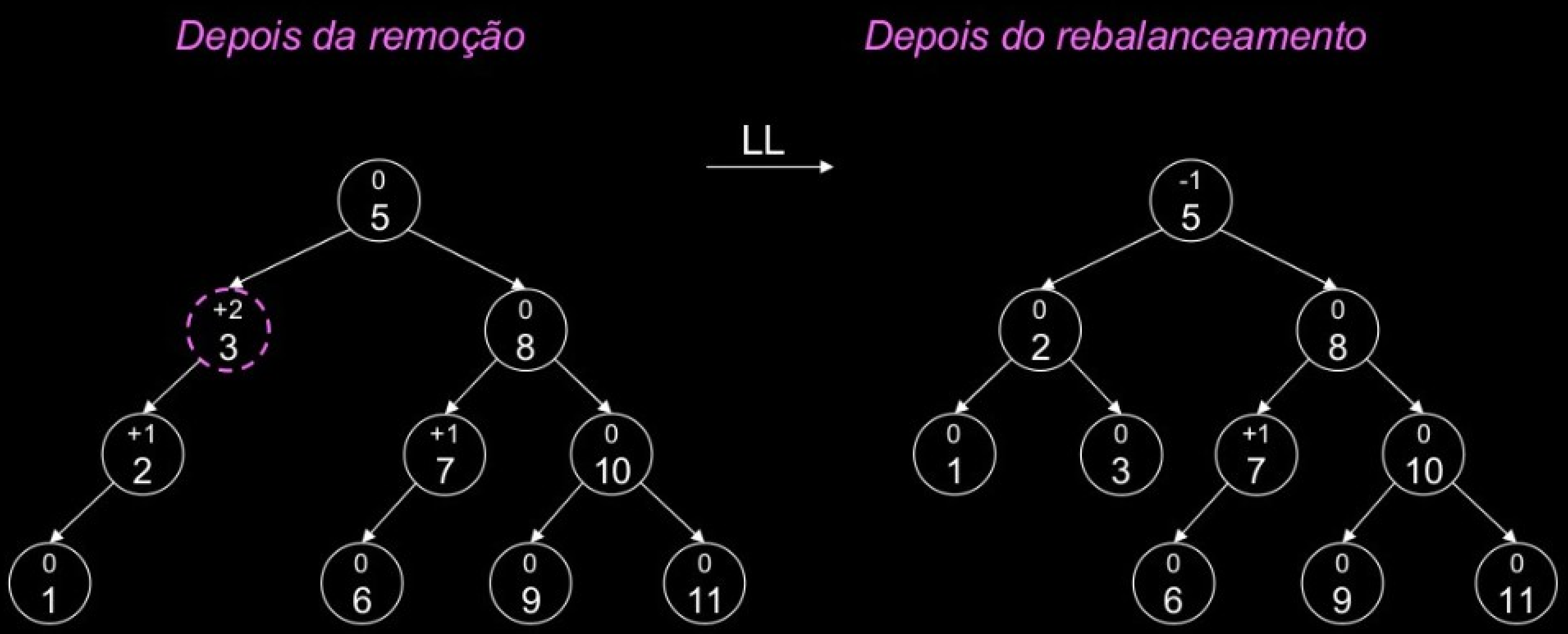
Antes da remoção



Remoção em árvores AVL

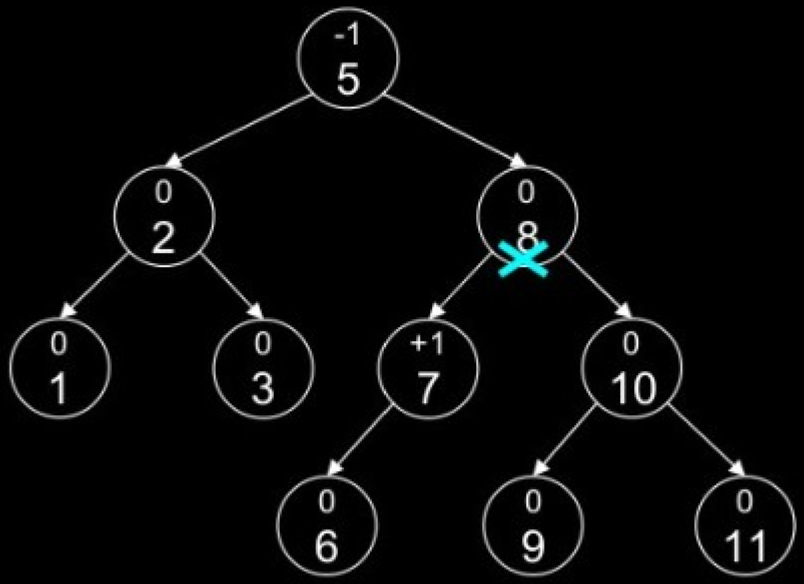


Remoção em árvores AVL



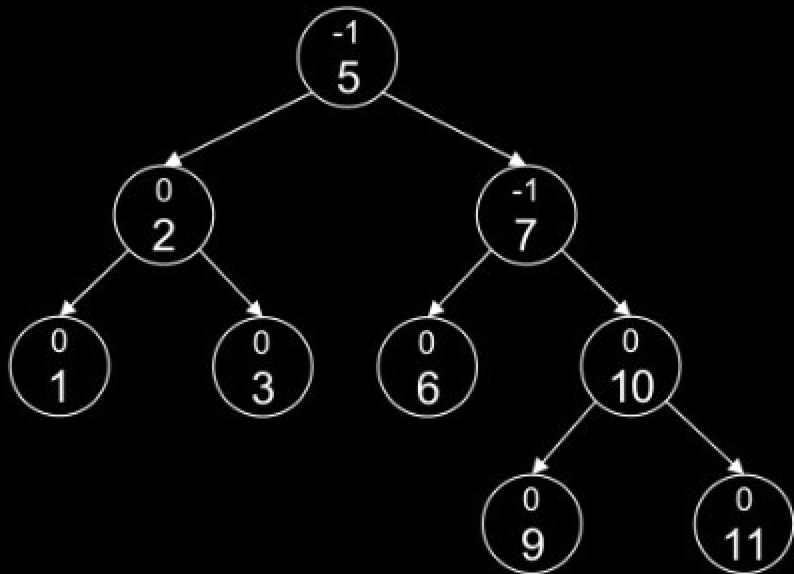
Remoção em árvores AVL

Antes da remoção



Remoção em árvores AVL

Depois da remoção



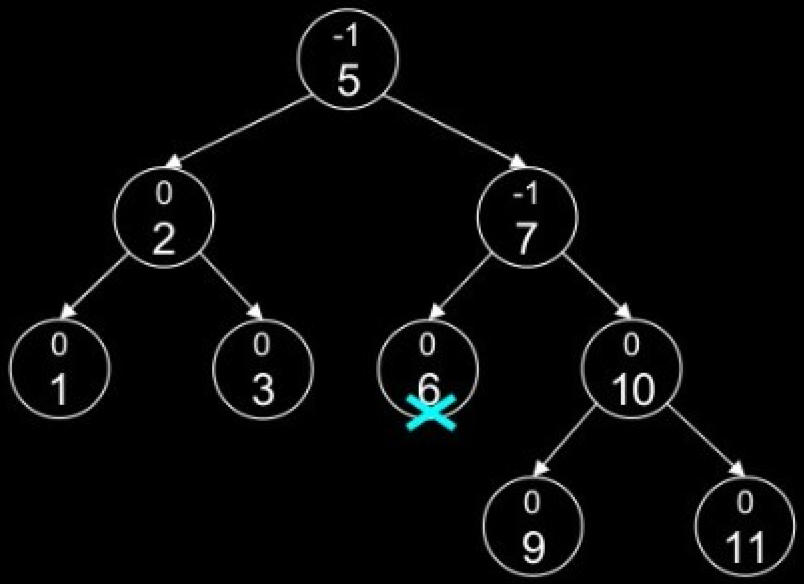
Depois do rebalanceamento

Sem necessidade
de rebalanceamento

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

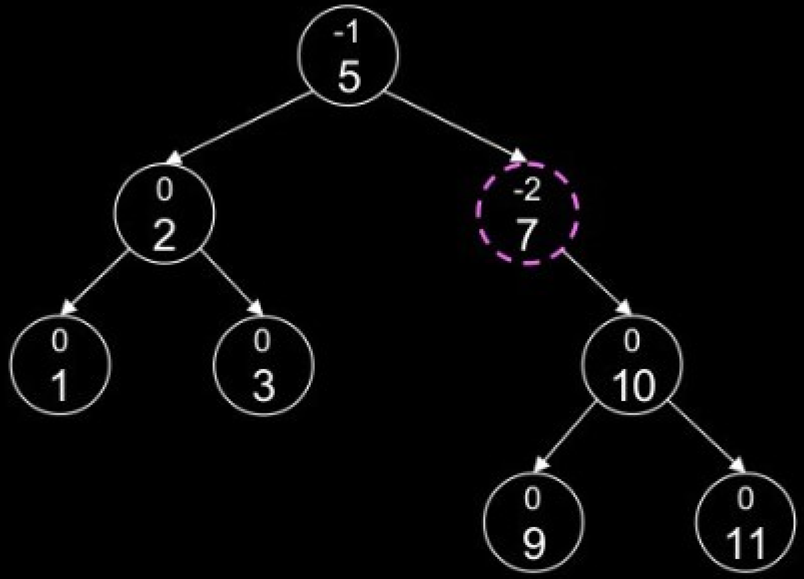
Remoção em árvores AVL

Antes da remoção

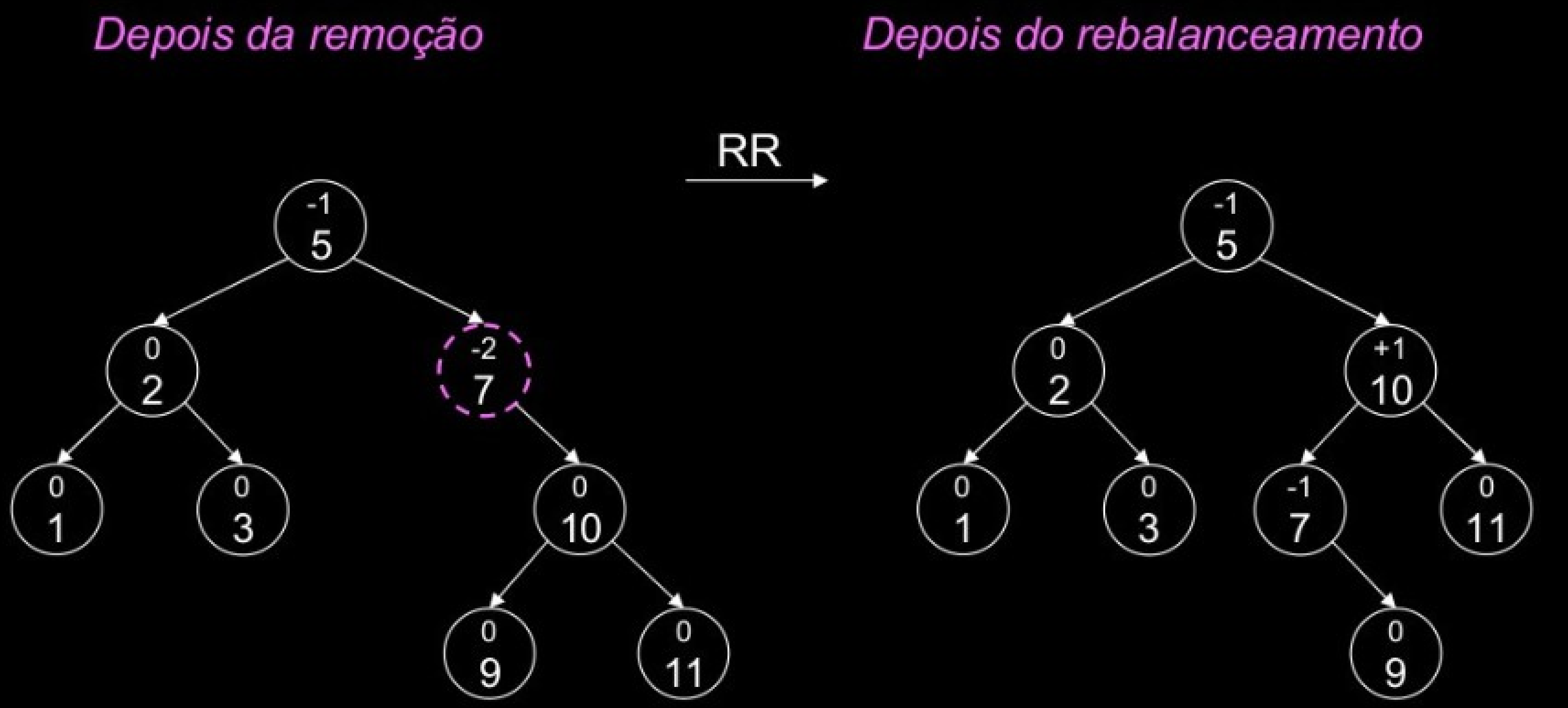


Remoção em árvores AVL

Depois da remoção

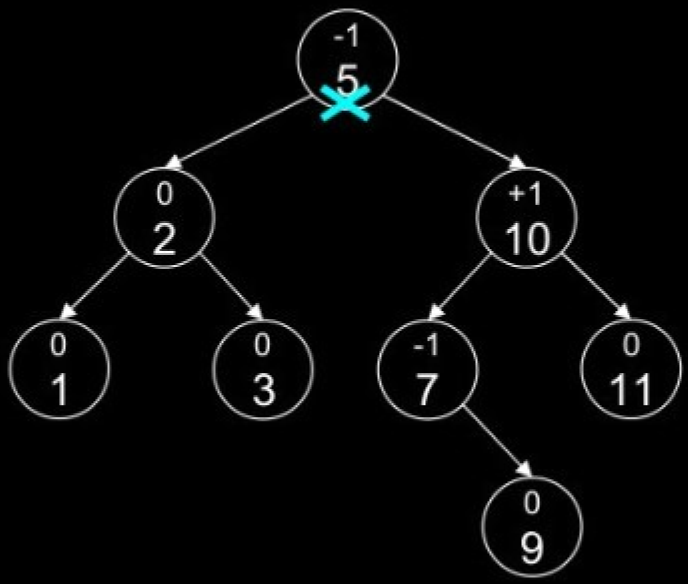


Remoção em árvores AVL



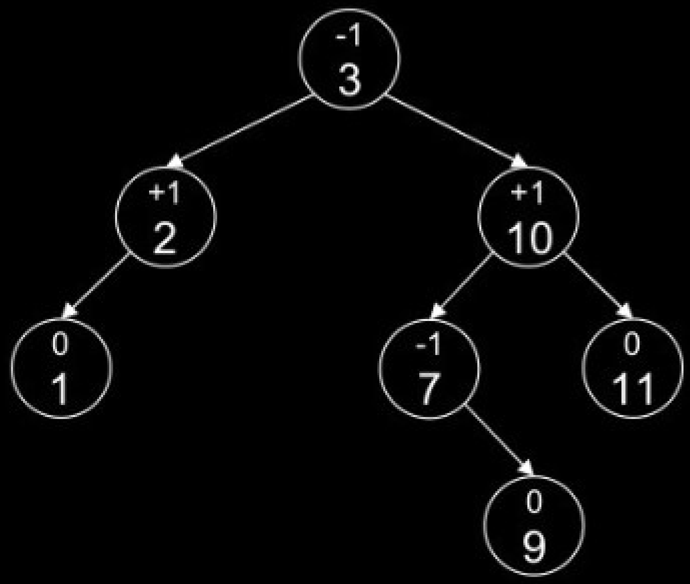
Remoção em árvores AVL

Antes da remoção



Remoção em árvores AVL

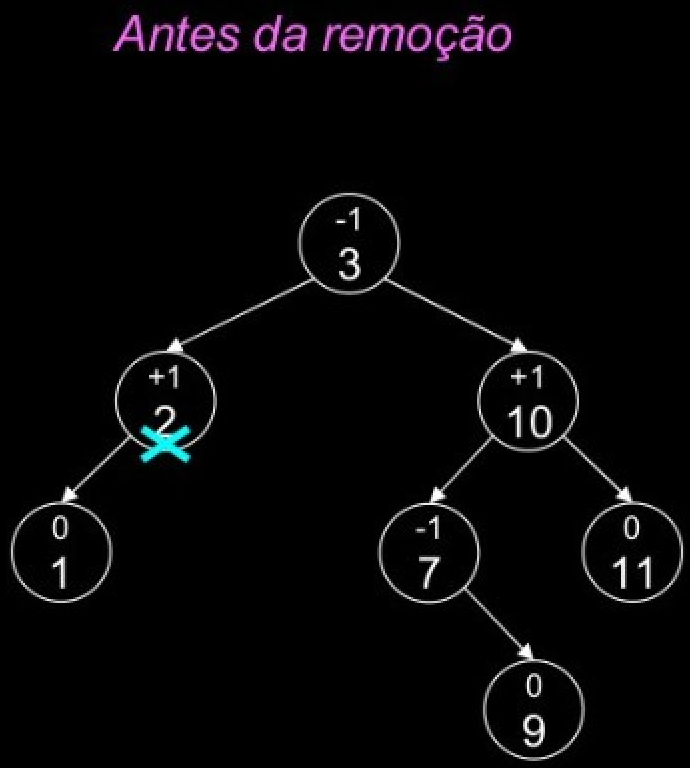
Depois da remoção



Depois do rebalanceamento

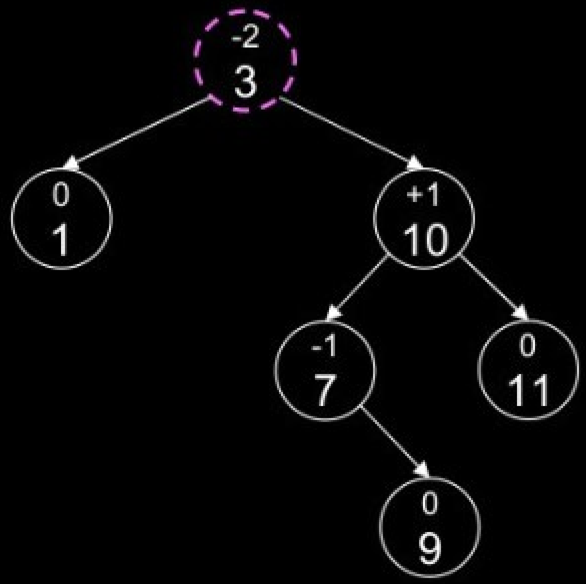
Sem necessidade de rebalanceamento

Remoção em árvores AVL

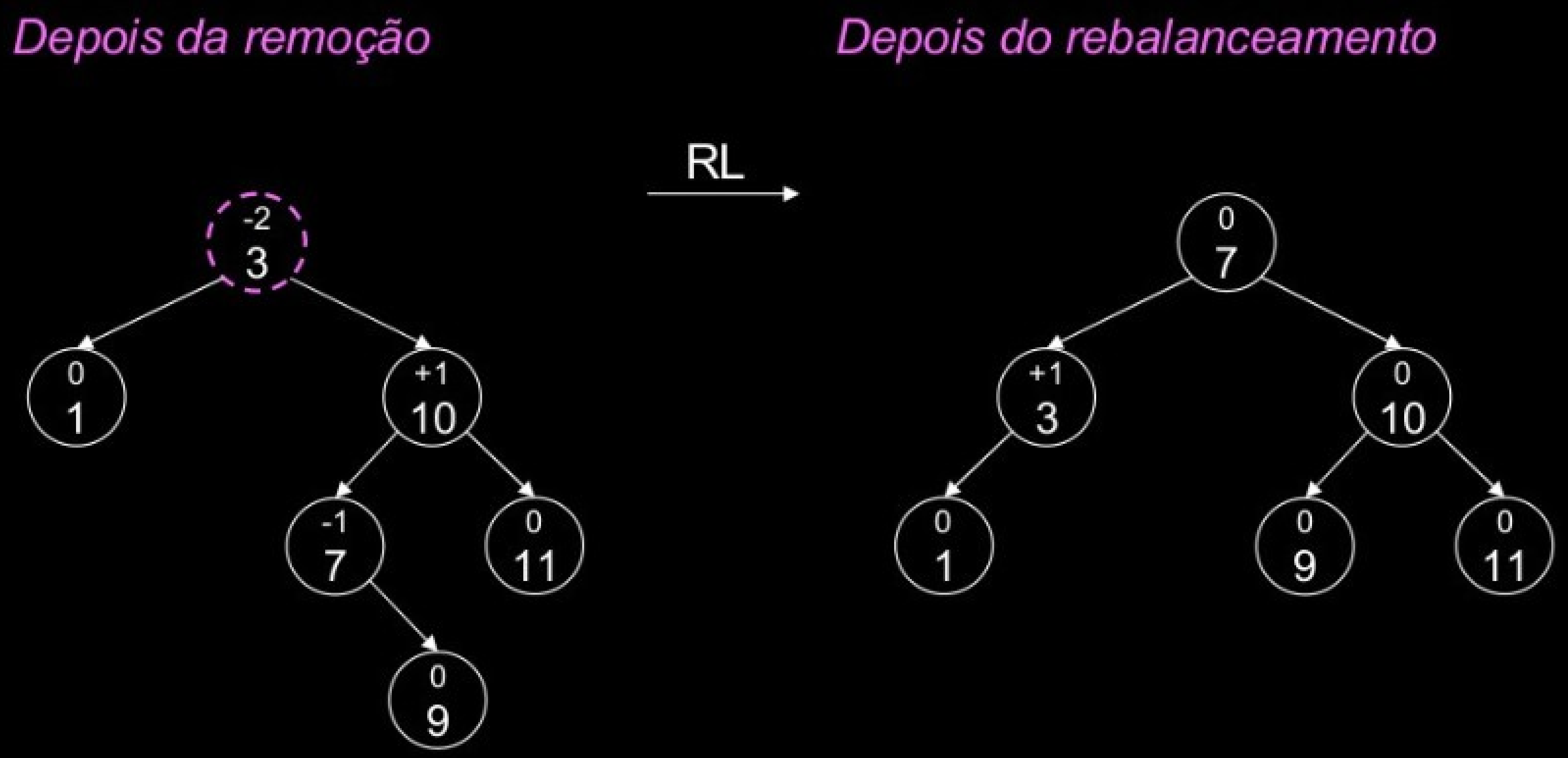


Remoção em árvores AVL

Depois da remoção



Remoção em árvores AVL



Referências:

Slides baseados em:

- Szwarcfiter, J.L. & Markezon, L. **Estruturas de Dados e seus Algoritmos**, 3a edição, LTC, 2010.
- Horowitz, E. & Sahni, S.; **Fundamentos de Estruturas de Dados**, Editora Campus, 1984.
- Wirth, N.; **Algoritmos e Estruturas de Dados**, Prentice/Hall do Brasil, 1989.
- Material de aula do **Prof. José Augusto Baranauskas** (USP/Riberão Preto)

AULA 19
ESTRUTURA DE DADOS

Árvores AVL: Complexidade e Remoção