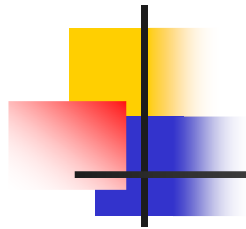


Árvores AVL (Adelson-Velskii and Landis)

Universidade Federal do Amazonas
Departamento de Eletrônica e Computação





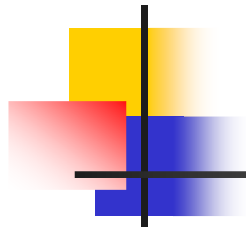
Introdução (1)

- **Árvore Balanceada**

- Uma árvore binária balanceada é aquela em que, para qualquer nó, suas sub-árvores esquerda e direita têm a mesma altura

- **Árvore AVL**

- Uma árvore binária de busca é balanceada quando, para cada nó, as alturas de suas subárvores (sa) esquerda e direita **diferem de, no máximo, 1**
- Essa diferença é chamada *fator de balanceamento*, ou $FB(n)$

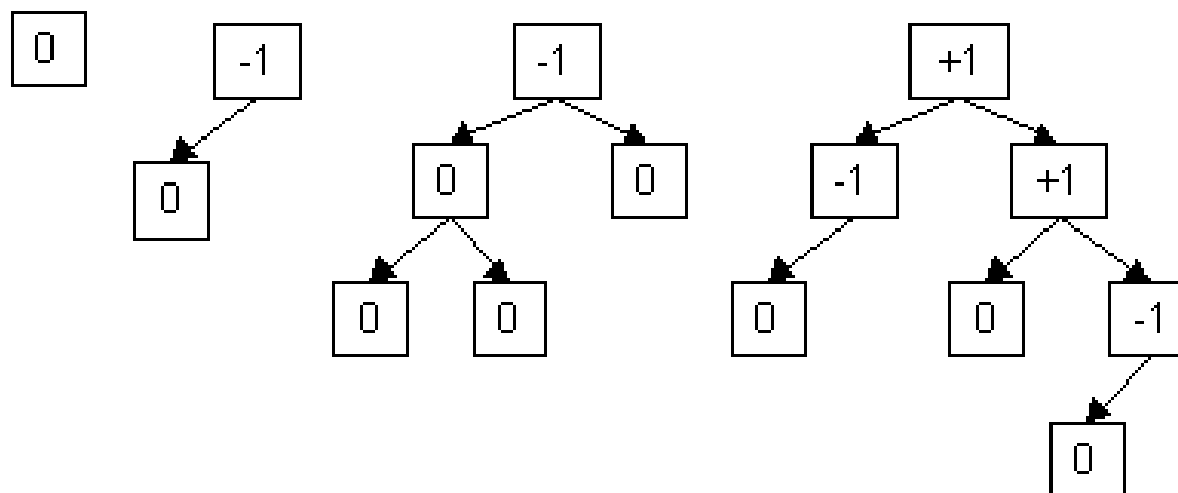


Introdução (2)

- Seja um nó **n** qualquer da árvore:
 - $FB(n) = altura(sad) - altura(sae)$
 - se $FB(n) = 0$, as duas sub-árvores têm a mesma altura
 - se $FB(n) = -1$, a sub-árvore esquerda é mais alta que a direita em 1
 - se $FB(n) = +1$, a sub-árvore direita é mais alta que a esquerda em 1



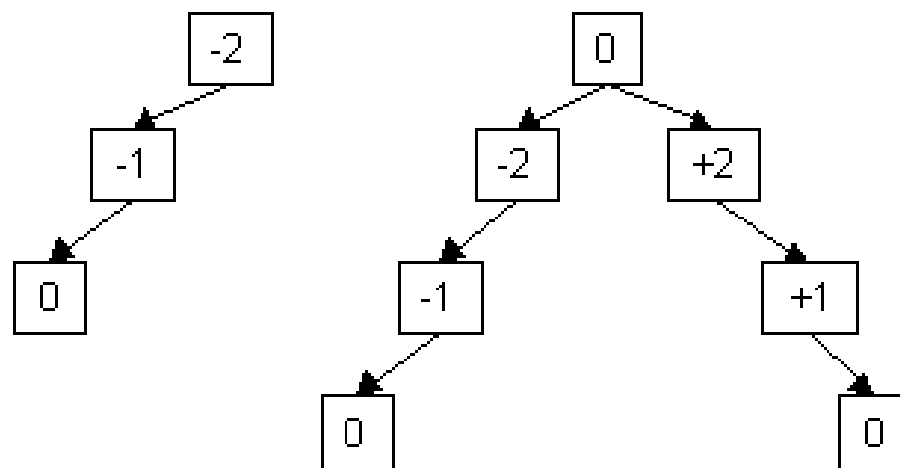
Introdução (3)



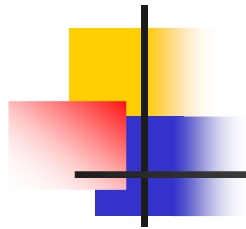
Exemplos de Árvores AVL



Introdução (4)

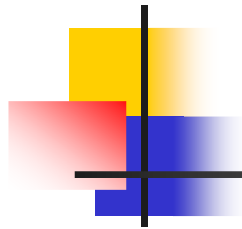


Exemplos de Árvores Não-AVL



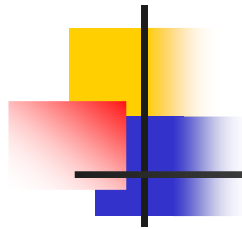
Introdução (5)

- A vantagem da árvore AVL sobre uma degenerada está na **eficiência** das suas operações de busca
 - Sendo a altura da AVL bem menor, o número necessário de comparações diminui sensivelmente
 - Numa árvore degenerada de 10.000 nós, são necessárias, em média, 5.000 comparações, numa busca; numa árvore AVL, com o mesmo número de nós, essa média baixa para 14
- O **algoritmo** deve, a cada **inserção**, fazer as **correções** necessárias para garantir que qualquer nó **n** tenha $|FB(n)| \leq 1$



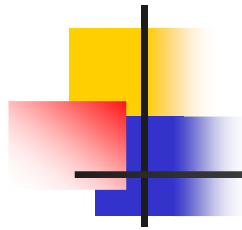
Balanceamento (1)

- Como fazemos então para manter uma árvore AVL balanceada?
 - Inicialmente inserimos um novo nodo na árvore normalmente
 - A inserção deste novo nodo **pode ou não** violar a propriedade de balanceamento
 - Caso a inserção do novo nodo **não viole** a propriedade de balanceamento podemos então continuar inserindo novos nodos
 - Caso contrário precisamos nos preocupar em **restaurar o balanço da árvore**
 - A restauração deste balanço é efetuada através do que denominamos de **rotações** na árvore



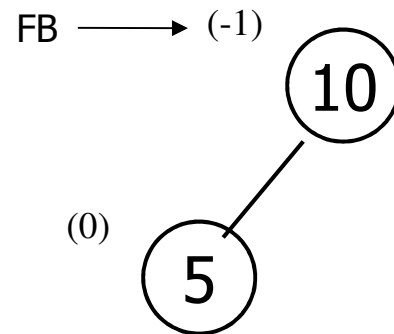
Balanceamento (2)

- Serão usados dois ponteiros **A** e **B**, para auxiliar:
 - **A** é nó ancestral **mais próximo** do nó inserido com $FB(nó) \neq 0$ antes da inserção
 - ou a própria raiz se não há nenhum nó com $FB(nó) \neq 0$ (antes da inserção) no *caminho da busca*
 - **A** é também chamado de **Pivô**
 - **B** é filho de **A** na sub-árvore onde ocorreu a inserção
 - Considerar **ligações unidirecionais** entre pai e filho

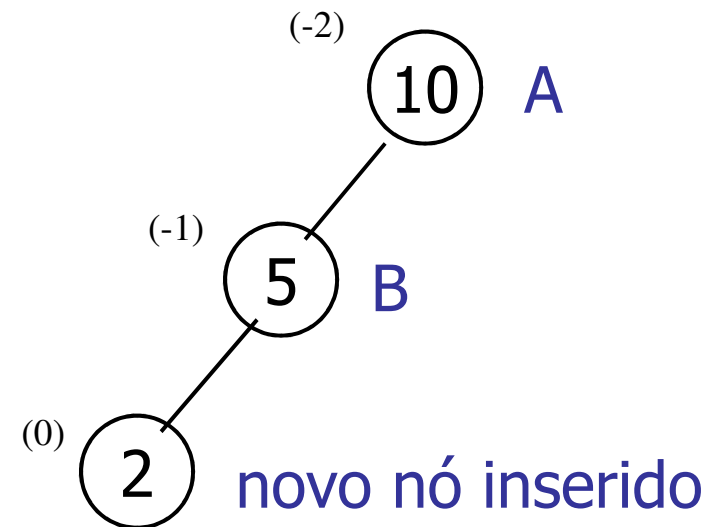


Exemplo de Desbalanceamento

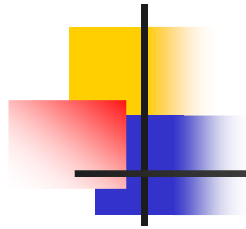
Antes da Inserção
do valor 2



Após a Inserção
do valor 2



Quem é A e quem é B?



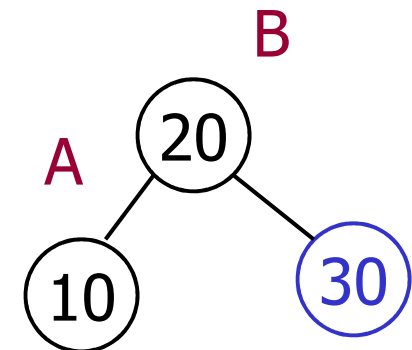
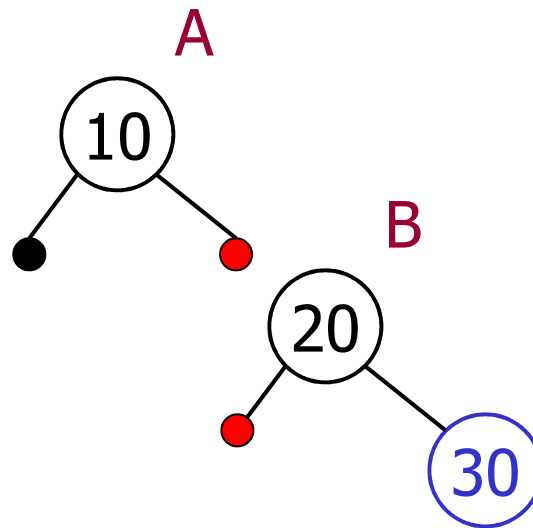
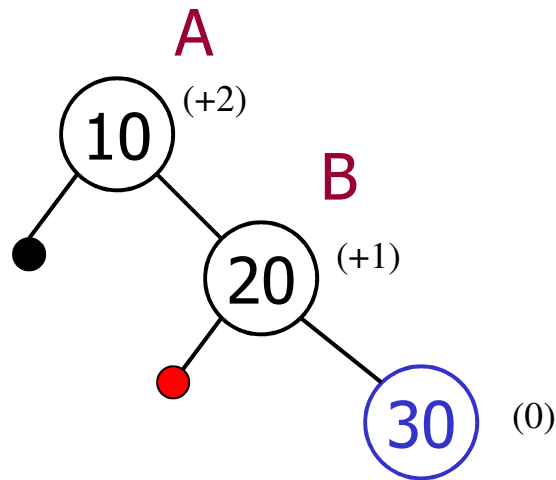
Rotação Simples

```
proc rotação simples
  se FB(A) = +1           // antes da inserção
    então rotação simples à esquerda
  senão rotação simples à direita
  fim se
  zera fatores de A e de B
fim proc
```

Rotação Simples à Esquerda

`A->dir = B->esq;`

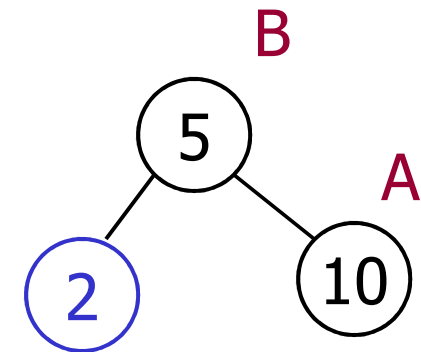
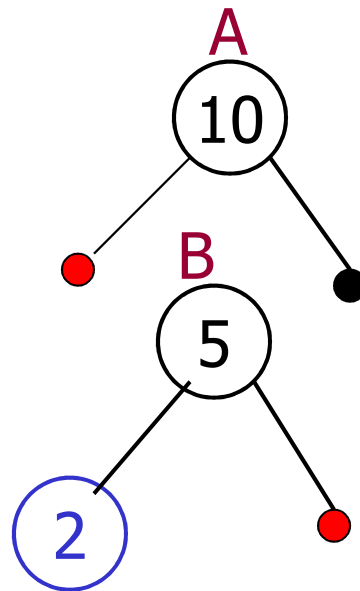
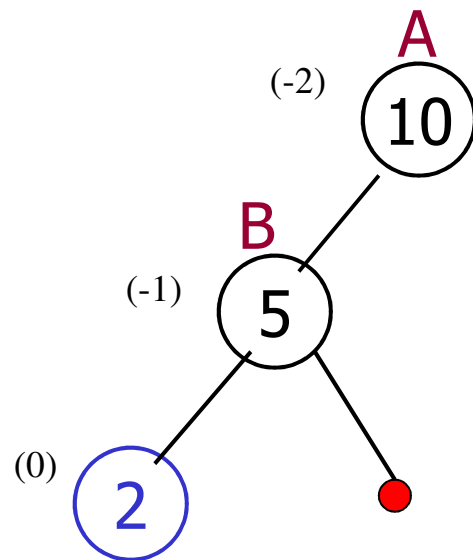
`B->esq = A;`



Rotação Simples à Direita

`A->esq = B->dir;`

`B->dir = A;`



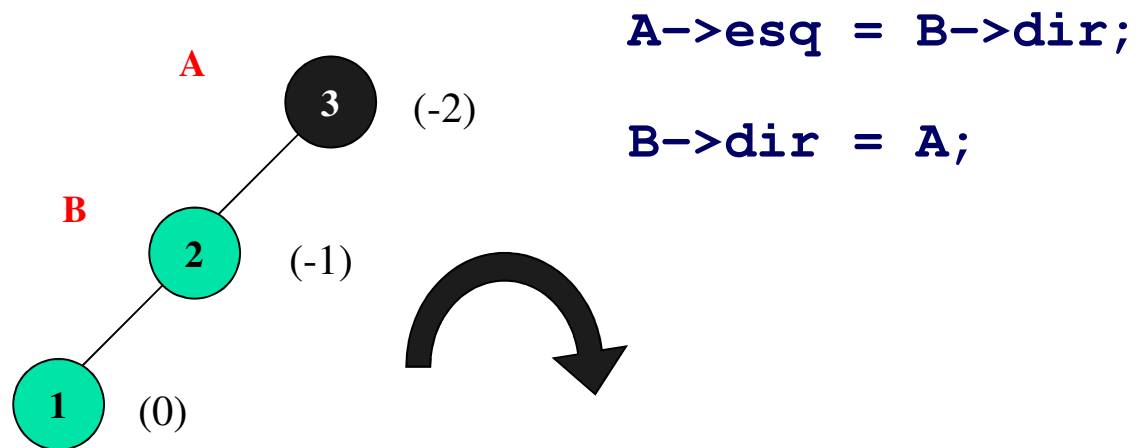


Exercício: Inserção (1)

- Mostrar as rotações necessárias para a construção da seguinte árvore AVL: 3, 2, 1, 4, 5, 6 e 7

Exercício: Inserção (2)

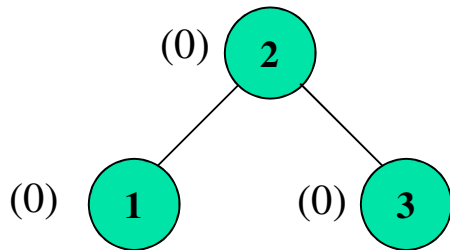
- Mostrar as rotações necessárias para a construção da seguinte árvore AVL: 3, 2, 1, 4, 5, 6 e 7



Quem é A? Quem é B? Quais os FB's?
O que é necessário fazer para equilibrar essa árvore?

Exercício: Inserção (3)

O resultado da rotação à direita fica...

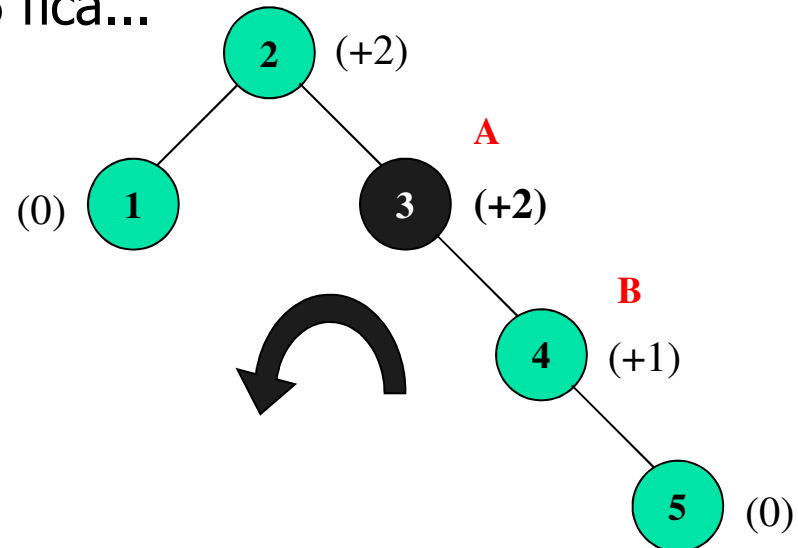


`B->pai = A->pai;`

`A->dir = B->esq;`

`B->esq = A;`

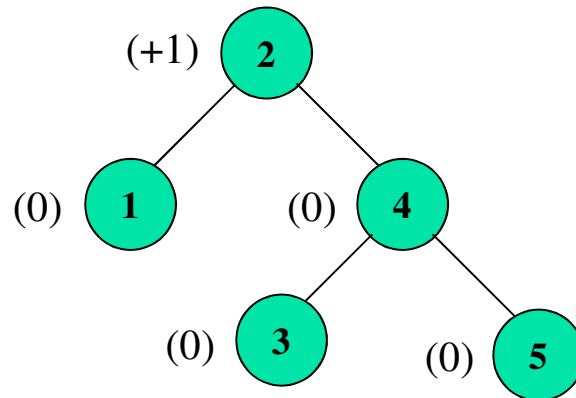
Após a inserção de 4 e 5 fica...



O que tem que ser feito para reequilibrar?

Exercício: Inserção (4)

O resultado da rotação à esquerda fica...

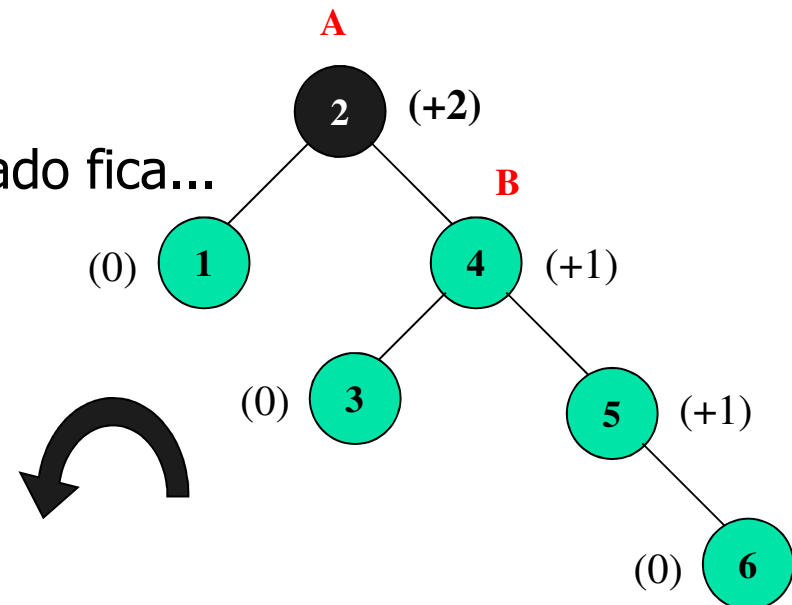


`B->pai = A->pai;`

`A->dir = B->esq;`

`B->esq = A;`

Mas quando o **6** é inserido o resultado fica...



O que tem que ser feito
para reequilibrar?

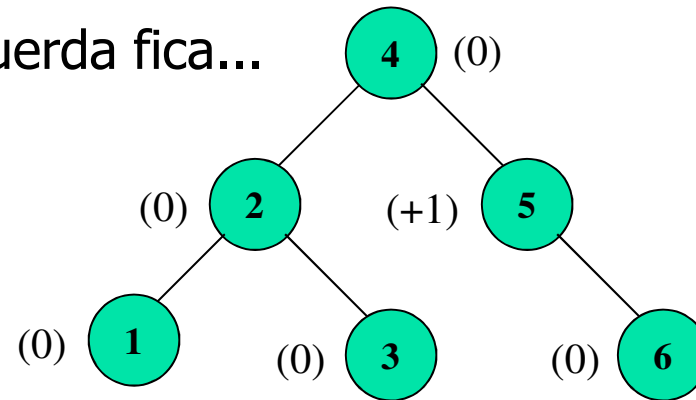
Exercício: Inserção (5)

O resultado da rotação à esquerda fica...

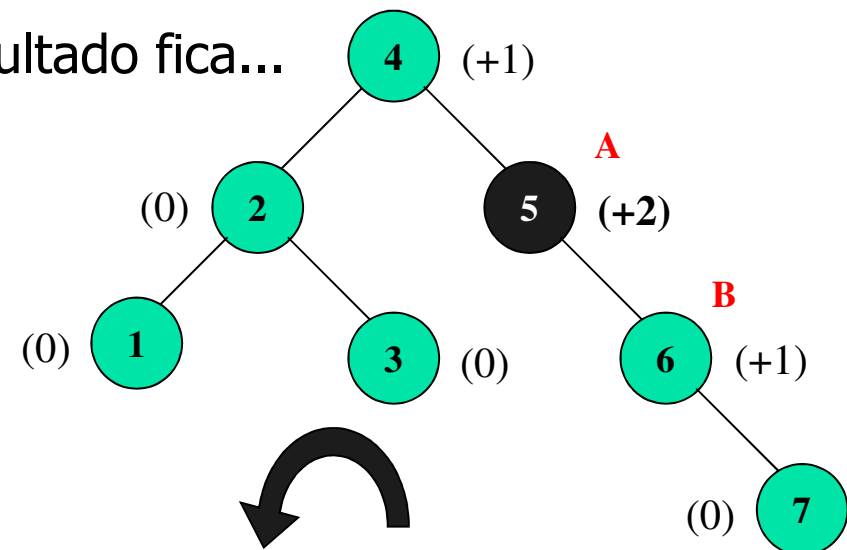
`B->pai = A->pai;`

`A->dir = B->esq;`

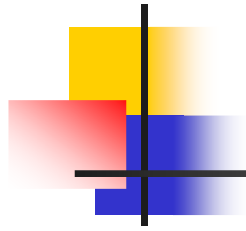
`B->esq = A;`



Mas quando o **7** é inserido o resultado fica...

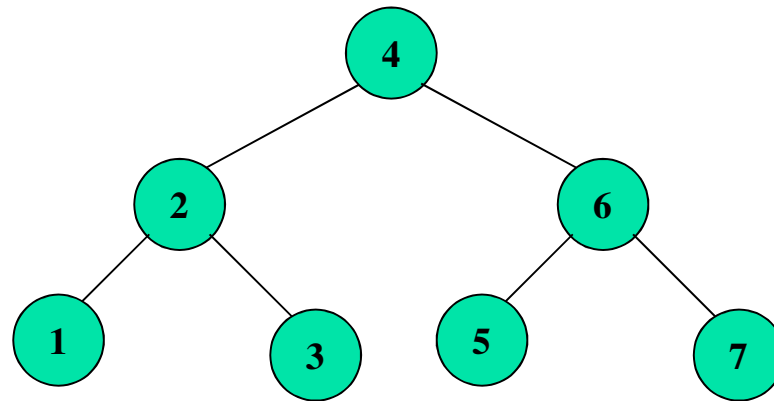


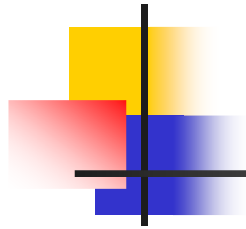
O que tem que ser feito para re-equilibrar?



Exercício: Inserção (6)

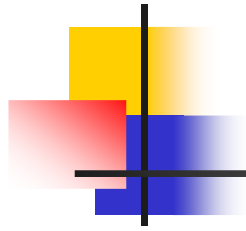
O resultado da rotação à esquerda fica...





Rotação Dupla

```
proc rotação dupla
  se FB(A) = +1           // antes da inserção
    então rotação dupla à direita
    senão rotação dupla à esquerda
  fim se
  ajusta fatores dos nós envolvidos na rotação
fim proc
```

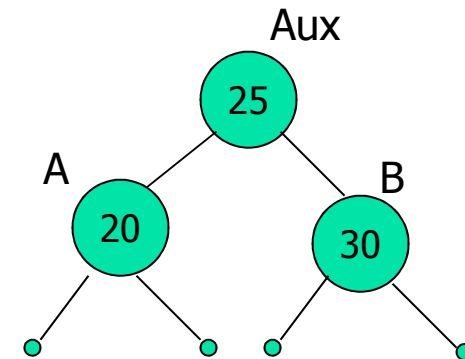
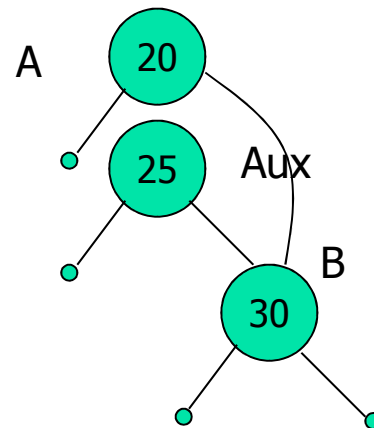
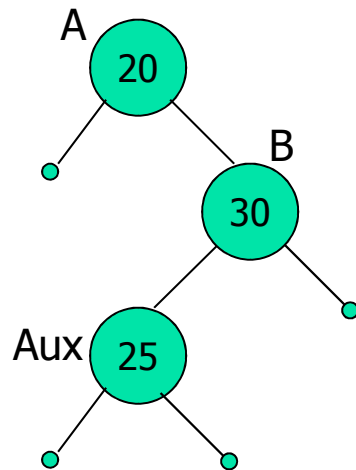


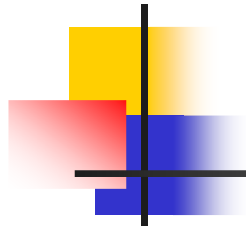
Rotação Dupla à Direita (1)

- É composta por uma rotação simples à **direita** (B e Aux) seguida de uma rotação simples à **esquerda** (A e Aux)
- Aux é o filho **esquerdo** de B

Rotação Dupla à Direita (2)

```
Aux = B->esq;  
// rotação simples à direita (B - Aux)  
B->esq = Aux->dir;  
Aux->dir = B;  
  
// rotação simples à esquerda (A - Aux)  
A->dir = Aux->esq;  
Aux->esq = A;
```



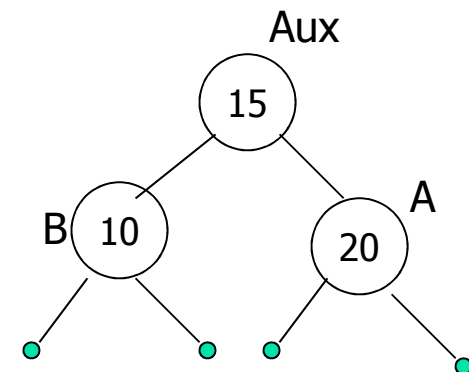
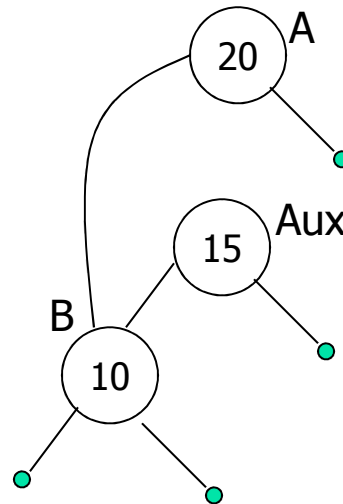
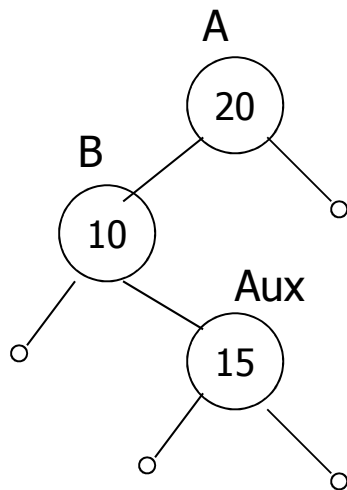


Rotação Dupla à Esquerda (1)

- É composta por uma rotação simples à **esquerda** (B e Aux) seguida de uma rotação simples à **direita** (A e Aux)
- Aux é o filho **direito** de B

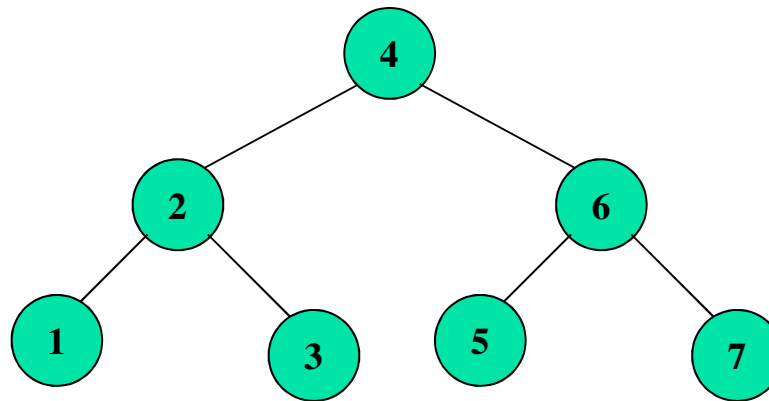
Rotação Dupla à Esquerda (2)

```
Aux = B->dir;  
// rotação simples à esquerda (B - Aux)  
B->dir = Aux->esq;  
Aux->esq = B;  
  
// rotação simples à direita (A - Aux)  
A->esq = Aux->dir;  
Aux->dir = A;
```



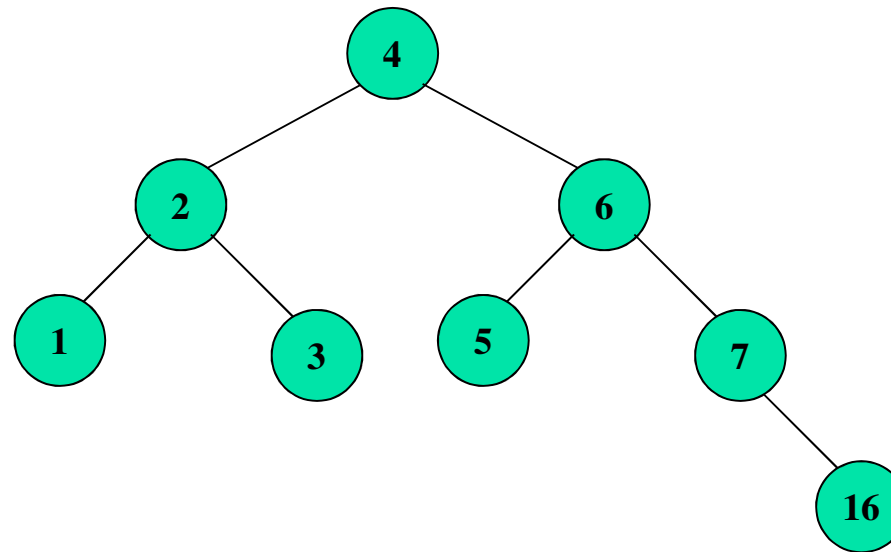


Exemplo: Rotação Dupla (1)



Como ficaria se fosse inserido o valor 16?

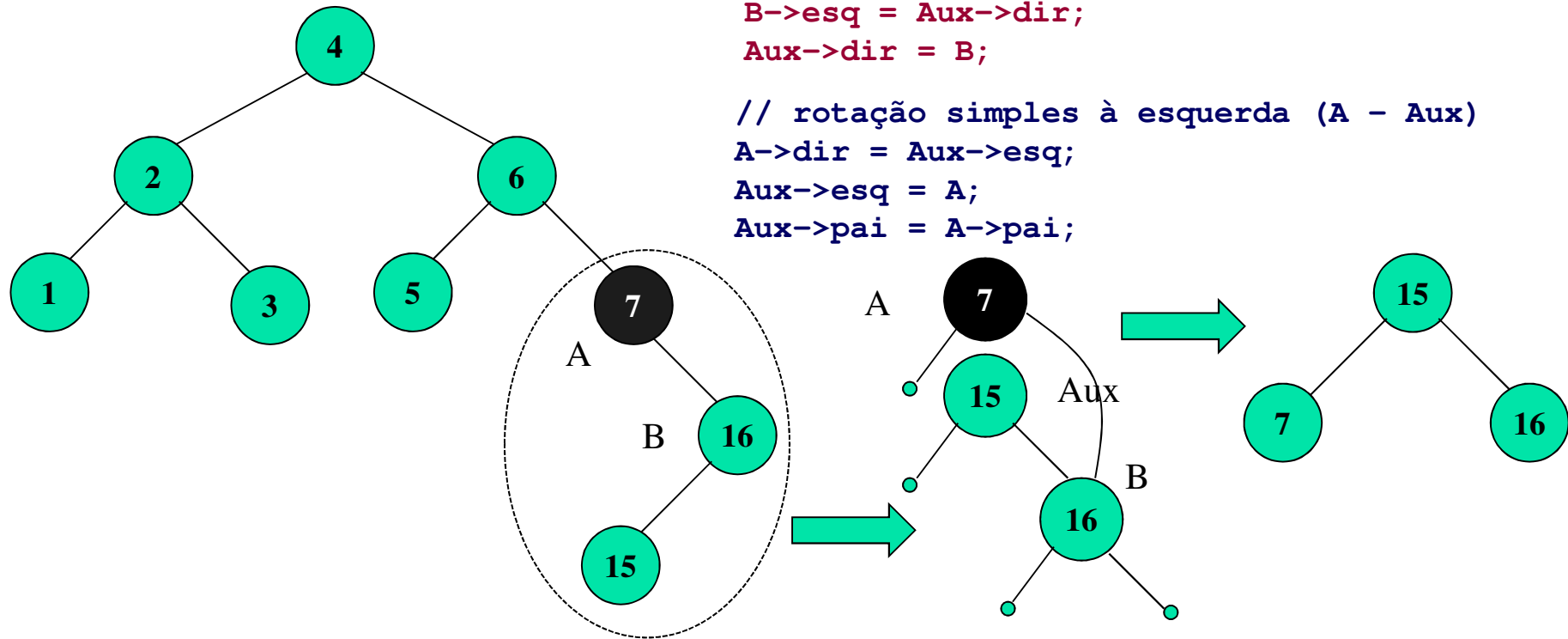
Exemplo: Rotação Dupla (2)



A Árvore ainda fica OK!

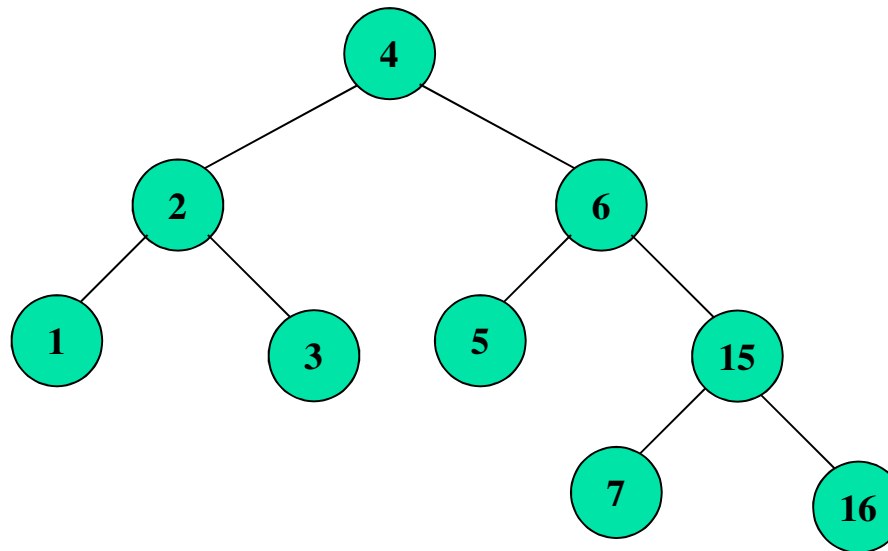
Como ficaria se fosse inserido o valor 15?

Exemplo: Rotação Dupla (3)



Desequilíbrio no nó 7. Rotação dupla à direita
Primeira fase: Rotação simples à direita
Segunda fase: Rotação simples à esquerda

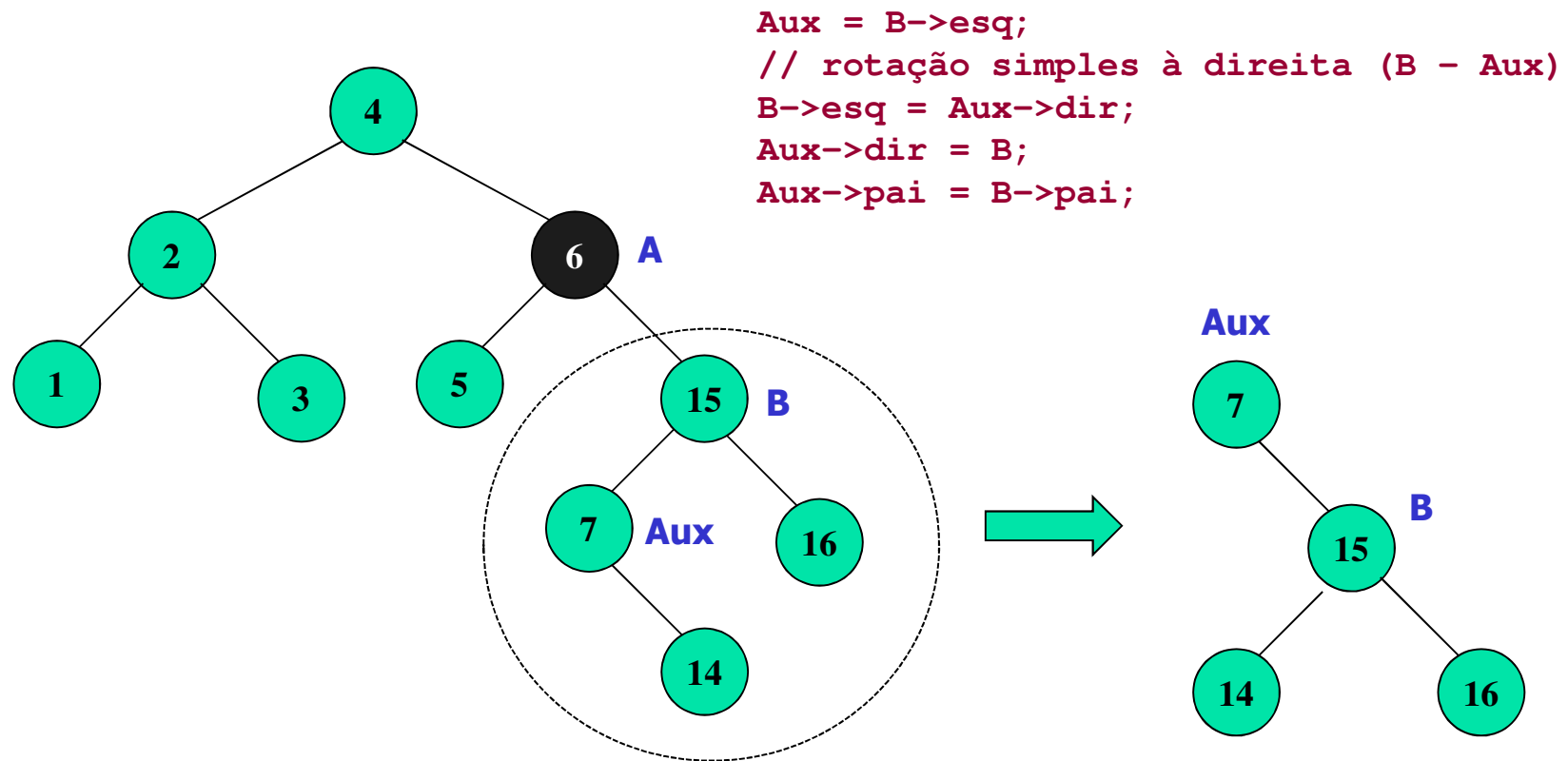
Exemplo: Rotação Dupla (4)



Agora a árvore está OK!

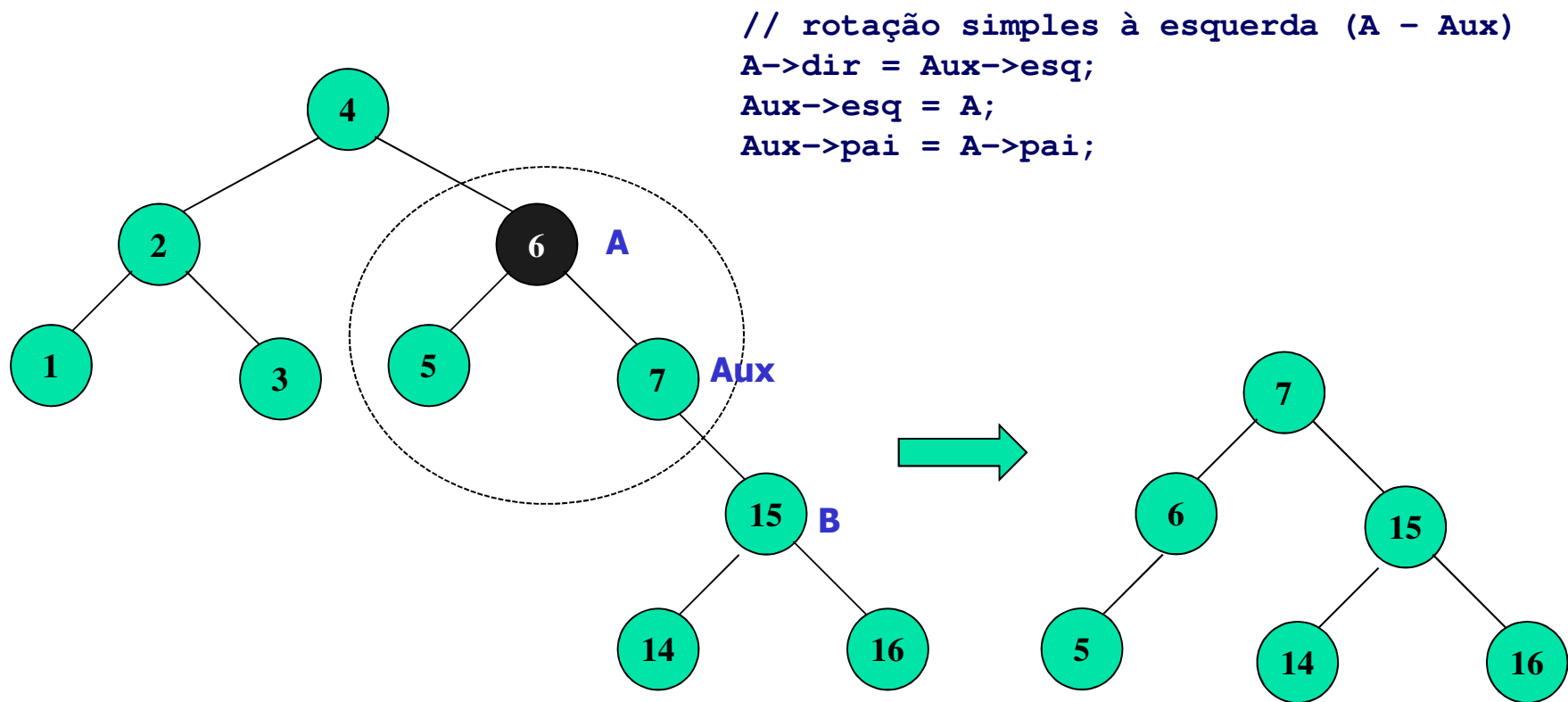
E se inseríssemos o 14?

Exemplo: Rotação Dupla (5)



Desequilíbrio no nó 6. Rotação dupla à direita
Primeira fase: Rotação simples à direita

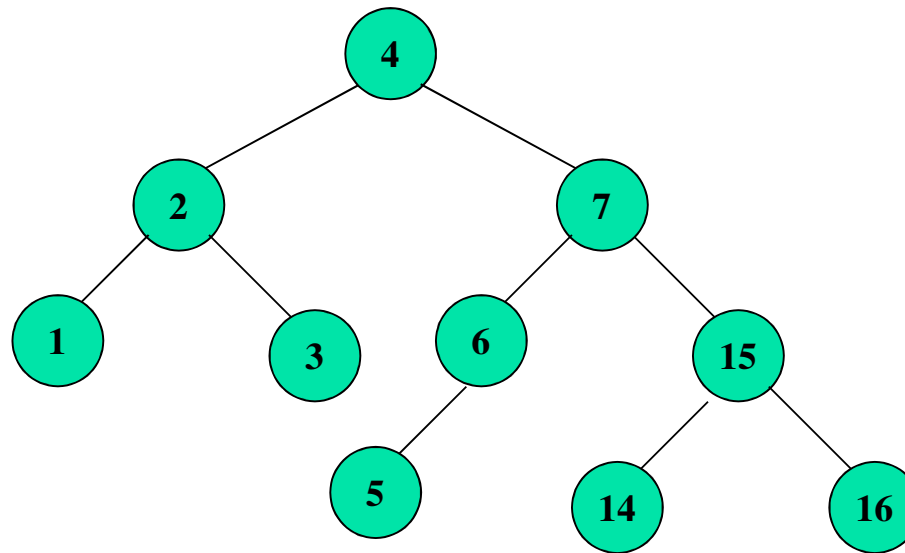
Exemplo: Rotação Dupla (6)



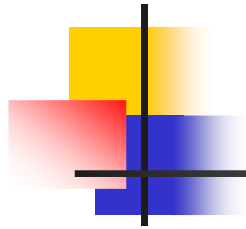
Segunda fase: Rotação simples à esquerda



Exemplo: Rotação Dupla (7)



Agora a árvore está OK!



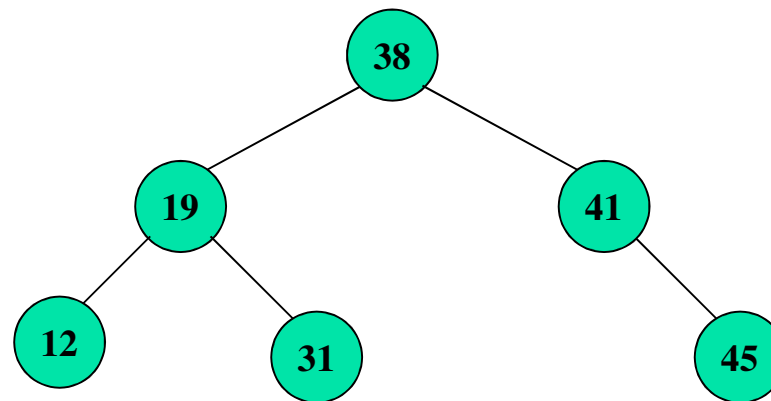
Exercício (1)

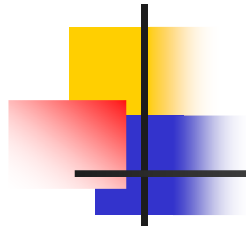
- Mostre a árvore AVL que resulta após a inserção bem-sucedida das chaves 41, 38, 31, 12, 19, 45

Exercício (1)

- Mostre a árvore AVL que resulta após a inserção bem-sucedida das chaves 41, 38, 31, 12, 19, 45

Resposta: Envolve uma rotação simples à direita e uma rotação dupla à esquerda





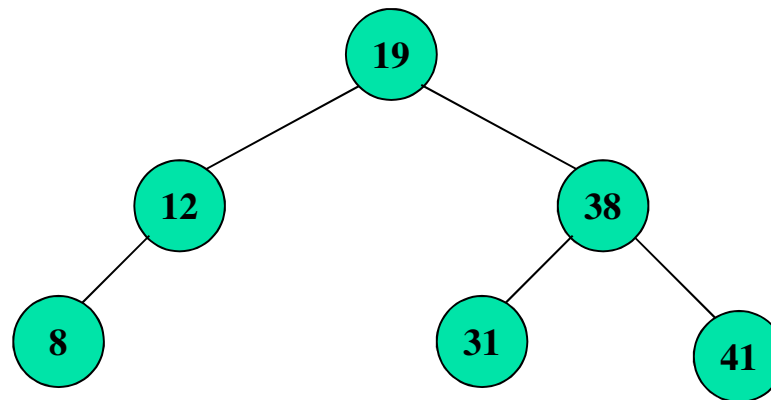
Exercício (2)

- Mostre a árvore AVL que resulta após a inserção bem-sucedida das chaves 41, 38, 31, 12, 19, 8

Exercício (2)

- Mostre a árvore AVL que resulta após a inserção bem-sucedida das chaves 41, 38, 31, 12, 19, 8

Resposta: Envole uma rotação simples à direita, uma rotação dupla à esquerda seguida de uma rotação simples à direita



<http://webdiis.unizar.es/asignaturas/EDA/AVLTree/avltree.html>



Remoção

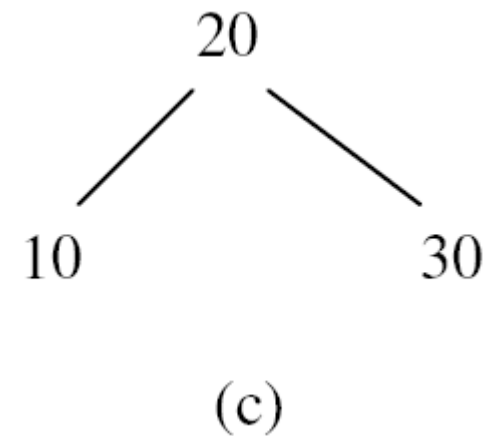
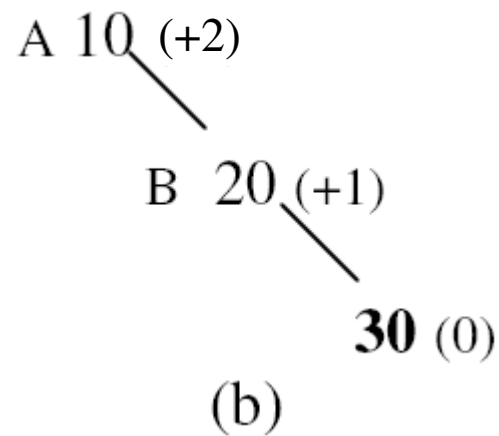
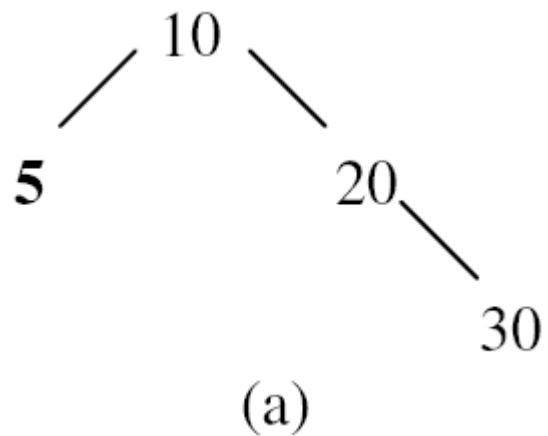
- Inicialmente, faz-se a retirada do nó, usando o algoritmo de busca e retirada de uma ABB
- Se não desbalanceou, o processo está encerrado
- Se desbalanceou a árvore, isto é, se um ou mais nós ficou com $|FB(nó)| > 1$, raciocina-se em termos de inserção, perguntando:
 - *se o desbalanceamento ocorresse devido a uma inserção, que nó teria sido inserido para causar tal desequilíbrio?*
- Identificado o nó, simula-se sua inserção e faz-se a rotação necessária

Remoção

Retirando o 5 resulta uma árvore **desbalanceada** no nó 10

Que nó inserido teria causado esse desequilíbrio? o 30

Uma **rotação simples à esquerda** resolve o problema

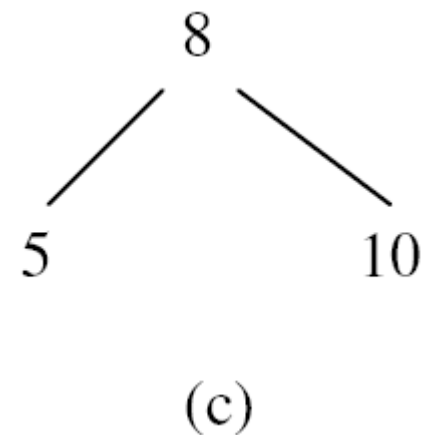
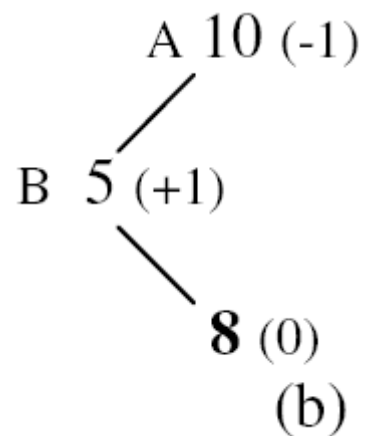
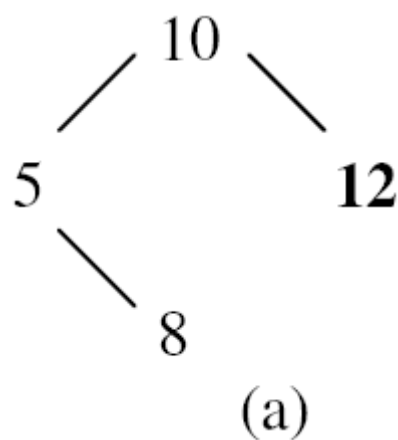


Remoção

Retirando o **12** desbalanceia a raiz

Podemos supor que a inserção recente foi o **8**

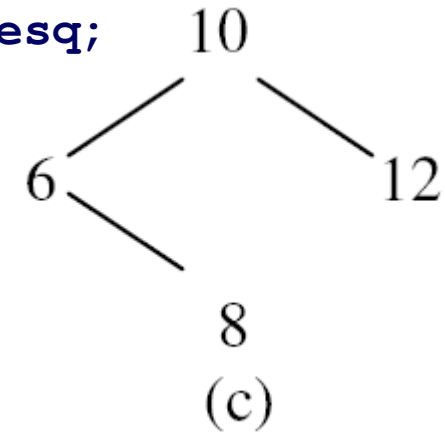
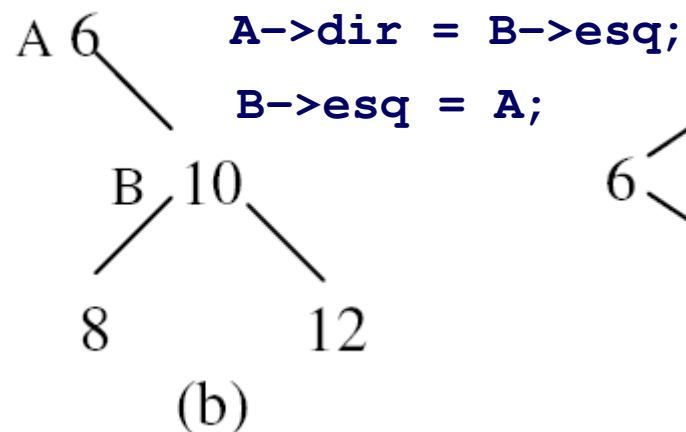
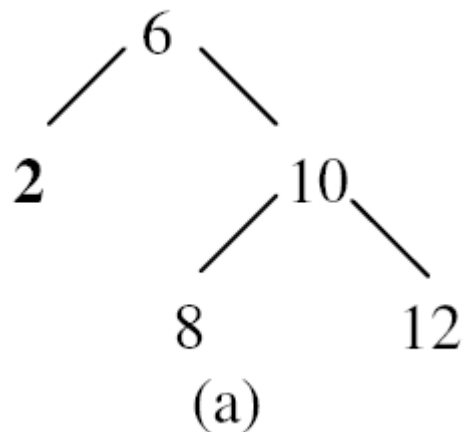
Uma **rotação dupla à esquerda** corrige o problema



Remoção

A retirada da folha 2 desbalanceia a raiz 6

Essa configuração jamais pode vir de uma seqüência de inserções, pois, se ela fosse 8, 12 ou 12, 8, a primeira dessas inclusões já provocaria rotação



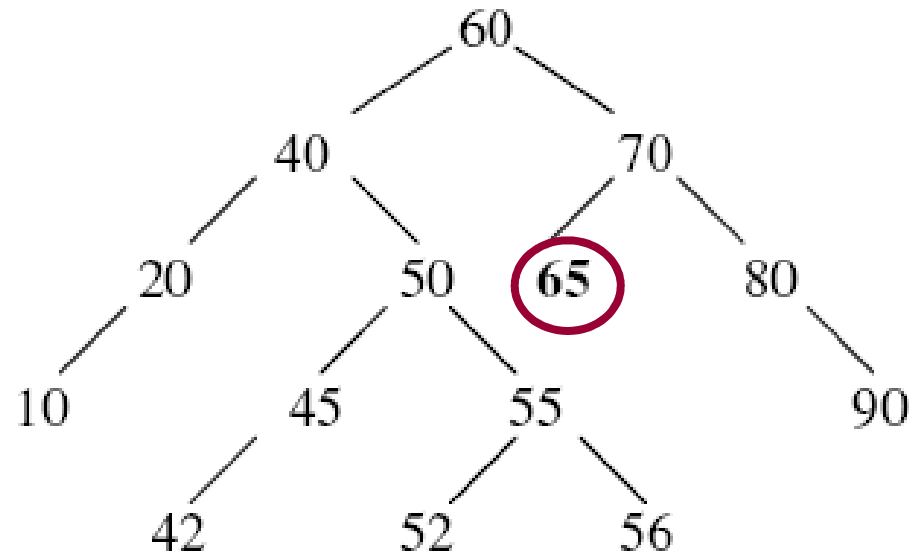
Solução: escolhe-se arbitrariamente um desses dois nós, despreza-se o outro (mantendo-o na árvore, obviamente), e simula-se a sua inserção

Escolhemos o 12, que exige uma operação mais simples: rotação simples à esquerda

Remoção

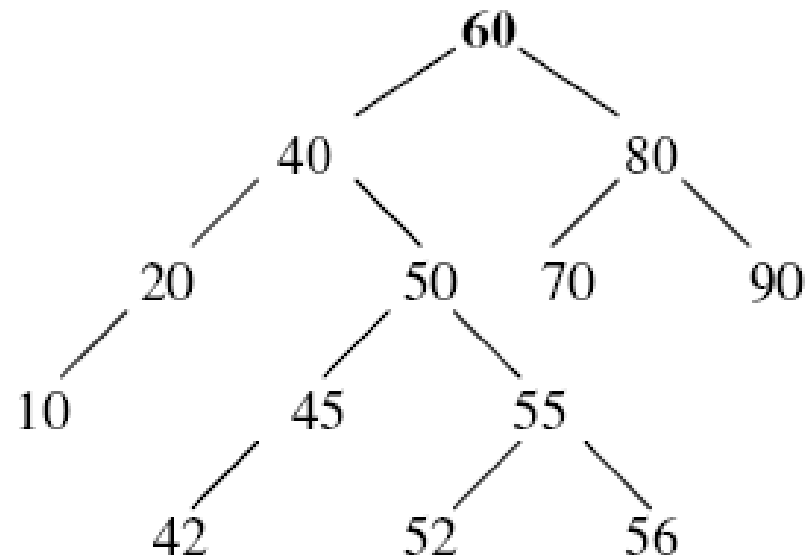
Infelizmente, há situações mais complexas, onde o próprio processo de balanceamento devido a retirada de um nó de uma subárvore, pode provocar um novo desequilíbrio na árvore

A solução será reaplicar o método para a árvore que desbalanceou. E novo desequilíbrio pode ser provocado mais acima, exigindo novo balanceamento. E assim por diante, até que toda a árvore volte a ser uma AVL



Remoção

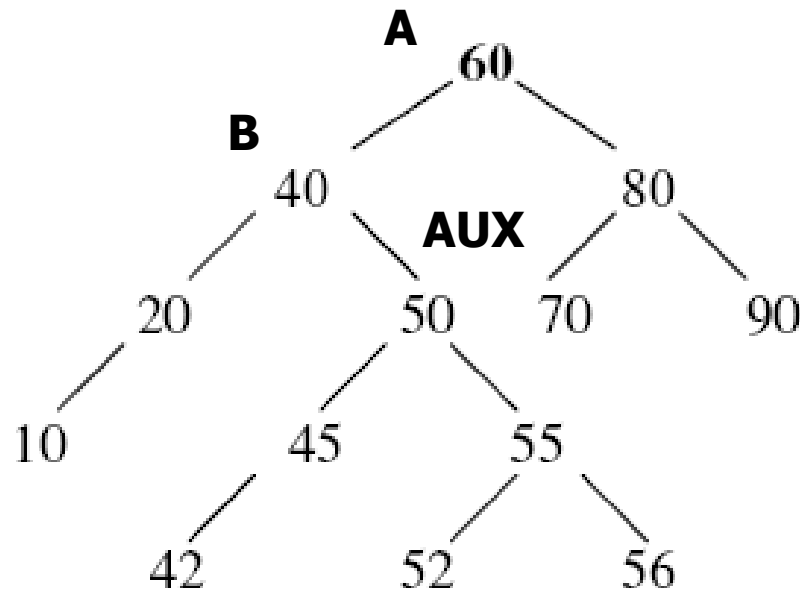
Isso causará o desequilíbrio da subárvore cuja raiz é **70**; aplicando nosso método para esta subárvore apenas, simulamos o ingresso do **90**, fazendo uma rotação simples à esquerda entre **70** e **80**, o que resulta na árvore abaixo:



Esta árvore é AVL?

Remoção

A árvore não é AVL, pois $|FB(60)| > 1$. Temos que reaplicar o método para o **60**. Considerando que, neste caso, tanto faz escolhermos o **42**, o **52** ou o **56** para ser o nó de inserção simulada, a rotação exigida é a dupla à esquerda (60-40), o resultado é a árvore abaixo que, finalmente, é uma AVL:

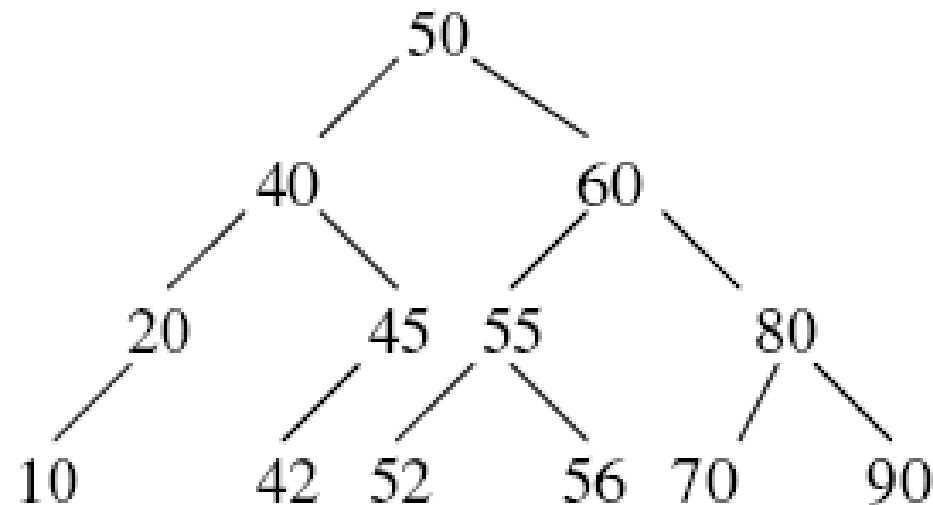


```
Aux = B->dir;
// rotação simples à esquerda (B - Aux)
B->dir = Aux->esq;
Aux->esq = B;
```

```
// rotação simples à direita (A - Aux)
A->esq = Aux->dir;
Aux->dir = A;
```

Remoção

A árvore não é AVL, pois $|FB(60)| > 1$. Temos que reaplicar o método para o 60. Considerando que, neste caso, tanto faz escolhermos o 42, o 52 ou o 56 para ser o nó de inserção simulada, a rotação exigida é a dupla à esquerda (60-40), o resultado é a árvore abaixo que, finalmente, é uma AVL:





Remoção

Isso mostra porque a remoção é mais complicada que a inserção. Enquanto nesta operação, no máximo uma rotação (simples ou dupla) servirá para manter a árvore balanceada, na remoção de um único nó, mais de uma rotação poderá ser necessária