

AVL

SCC-202 – Algoritmos e Estruturas de
Dados I

Prof. Thiago A. S. Pardo

Árvores binárias de busca (ABB)

- Muito boas para busca
 - $O(\log n)$
- Sabe-se que
 - **Lista encadeada**
 - Eficiente para **inserção** e **remoção** dinâmica de elementos, mas ineficiente para busca
 - **Lista sequencial (ordenada)**
 - Eficiente para **busca**, mas ineficiente para inserção e remoção de elementos

mas... **ABBs**: solução eficiente para inserção,
remoção e busca

ABB

- Contra-exemplo
 - Inserção dos elementos na ordem em que aparecem
 - A, B, C, D, E, ..., Z
 - 1000, 999, 998, ..., 1

3

ABB

- O **desbalanceamento** da árvore pode tornar a busca tão ineficiente quanto a busca sequencial (no pior caso)
 - **$O(N)$**
- Solução?

4

ABB

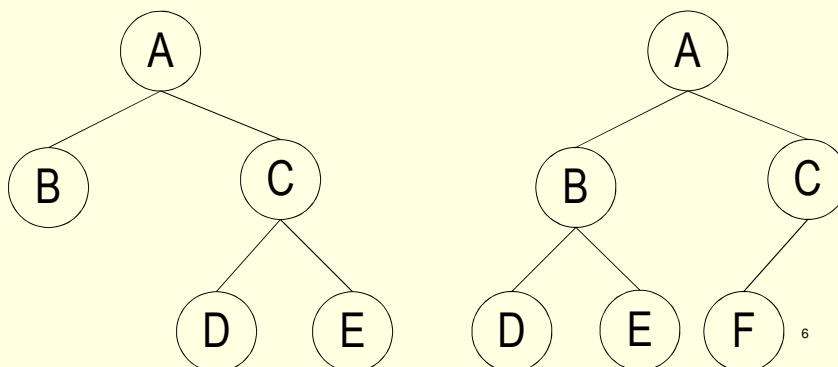
- O **desbalanceamento** da árvore pode tornar a busca tão ineficiente quanto a busca sequencial (no pior caso)
 - **$O(N)$**
- Solução?

Balanceamento da árvore!

5

Árvores balanceadas

- Uma árvore binária é dita balanceada se, para cada nó, as alturas de suas duas subárvores diferem de, no máximo, 1



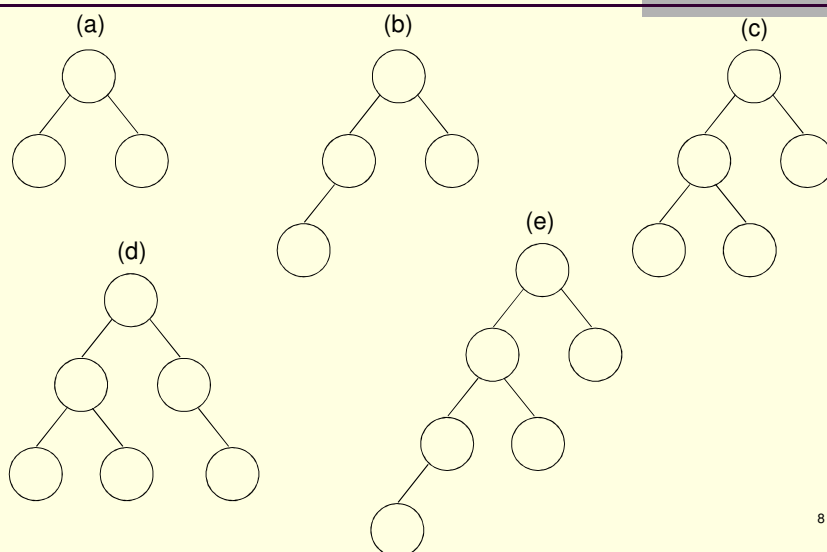
6

AVL

- Árvore binária de busca balanceada
 - Para cada nó, as alturas das subárvores diferem em 1, no máximo
 - Proposta em 1962 pelos matemáticos russos G.M. Adelson-Velski e E.M. Landis
 - Métodos de **inserção** e **remoção** de elementos da árvore de forma que ela fique balanceada

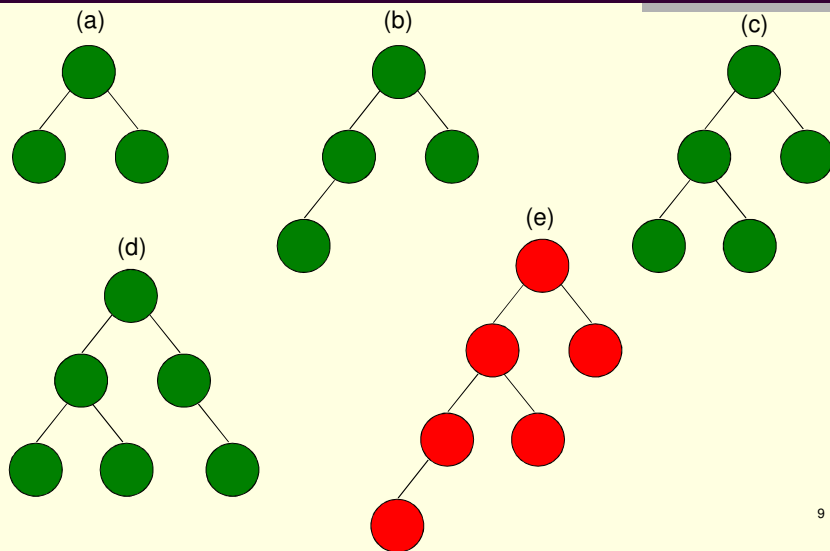
7

AVL: quem é e quem não é?

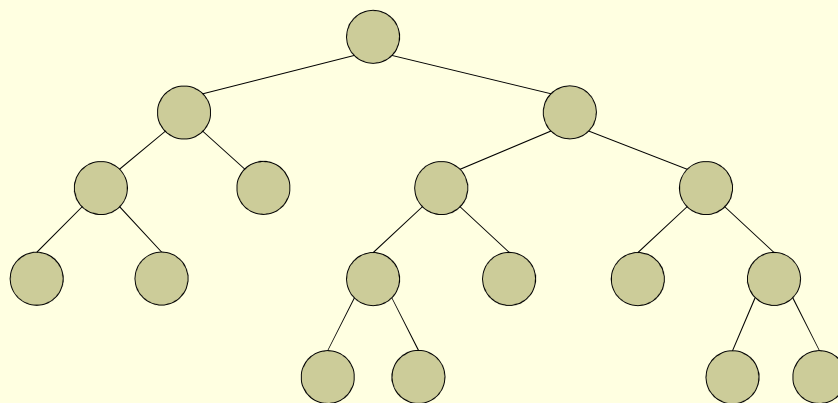


8

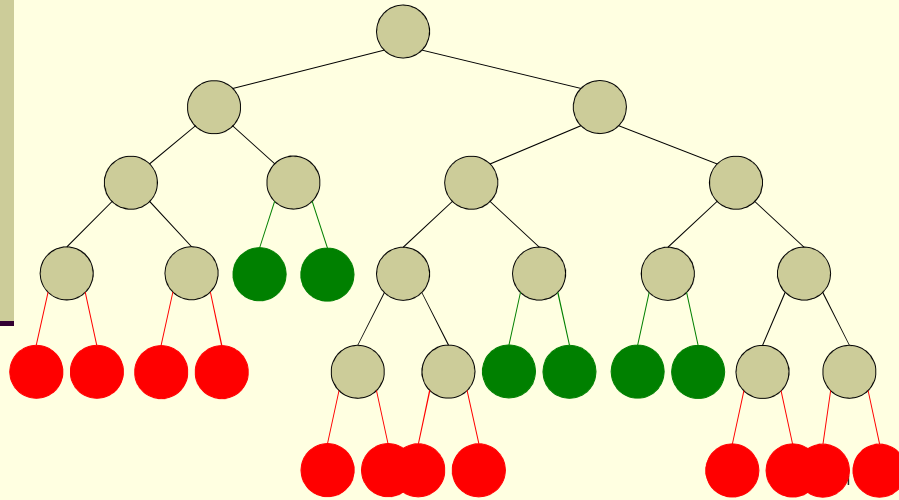
AVL: quem é e quem não é?



Pergunta: a árvore abaixo é AVL?



Exercício: onde se pode incluir um nó para a AVL continuar sendo AVL?



AVL

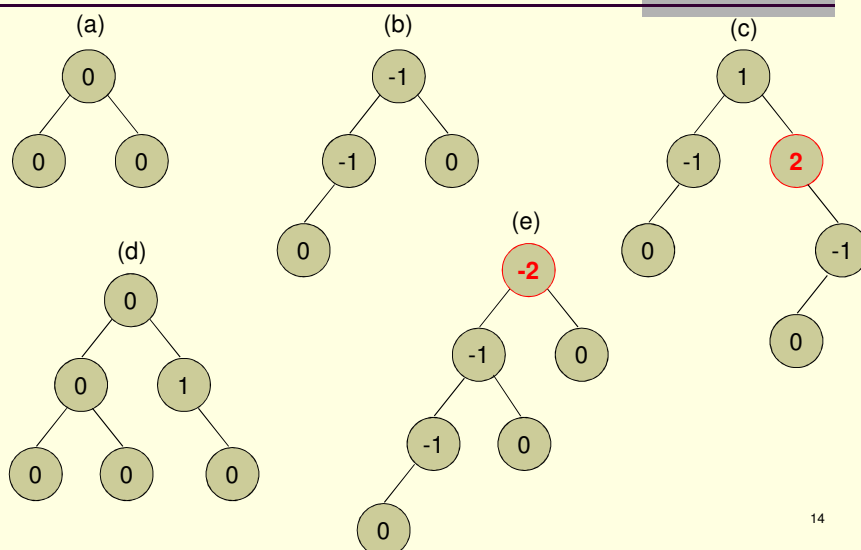
- Como é que se sabe **quando é necessário balancear** a árvore?
 - Se a diferença de altura das subárvores deve ser 1, no máximo, então temos que procurar diferenças de altura maior do que isso
 - Possível solução: cada nó pode manter a diferença de altura de suas subárvores
 - Convencionalmente chamada de **fator de balanceamento** do nó

AVL

- **Fatores de balanceamento** dos nós
 - Altura da subárvore direita menos altura da subárvore esquerda
 - Hd-He
 - Atualizados sempre que a árvore é alterada (elemento é inserido ou removido)
 - Quando um fator é 0, 1 ou -1, a árvore está balanceada
 - Quando um fator se torna 2 ou -2, a árvore está desbalanceada
 - Operações de balanceamento!

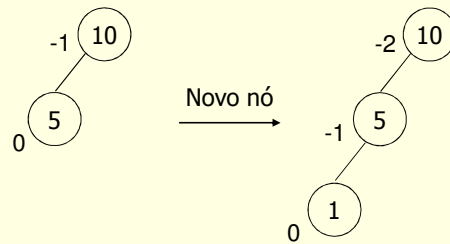
13

AVL: quem é e quem não é



14

AVL: exemplo de desbalanceamento

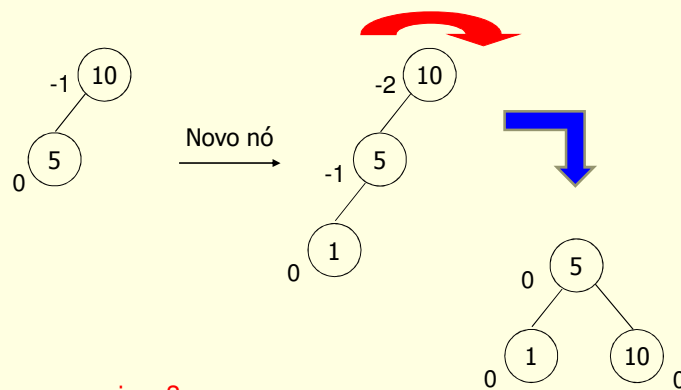


Desbalanceou!!!

Como arrumar isso?

15

AVL: exemplo de desbalanceamento

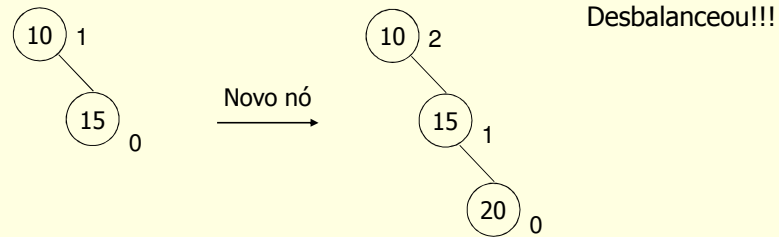


Como arrumar isso?

Rotação simples para direita!

16

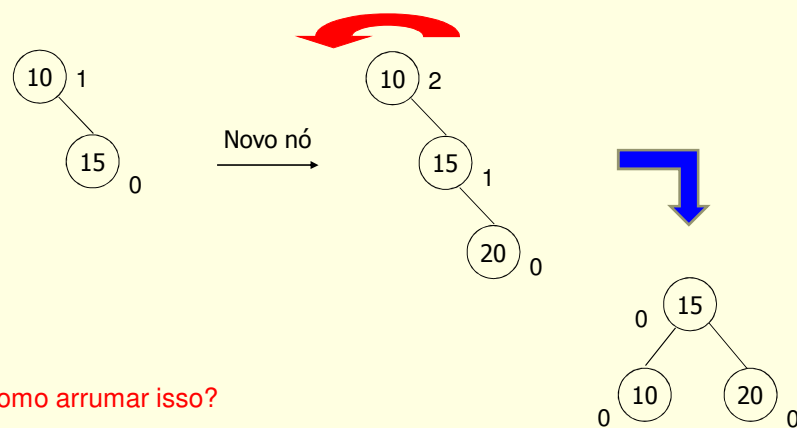
AVL: exemplo de desbalanceamento



Como arrumar isso?

17

AVL: exemplo de desbalanceamento

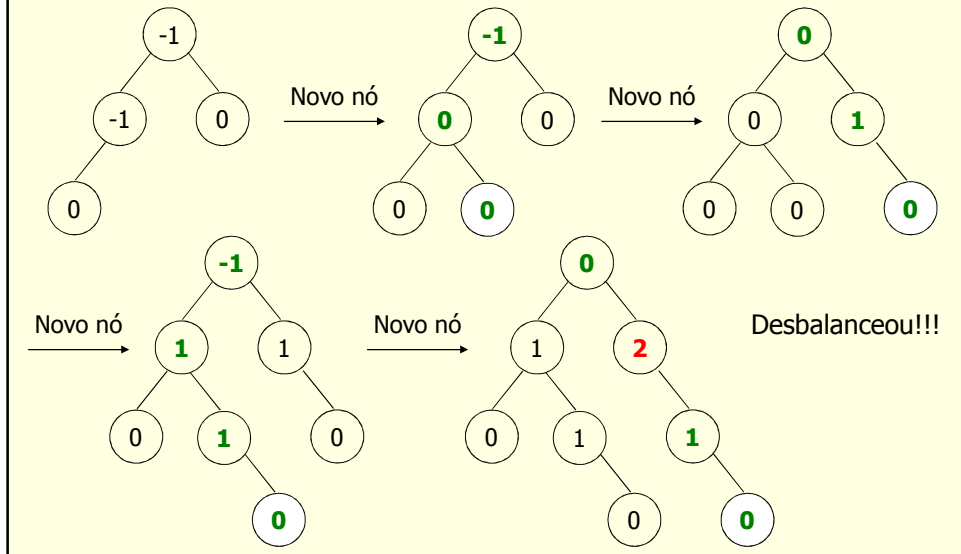


Como arrumar isso?

Rotação simples para esquerda!

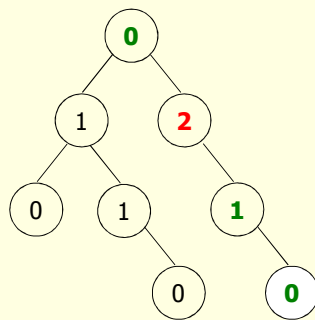
18

AVL: exemplo de desbalanceamento



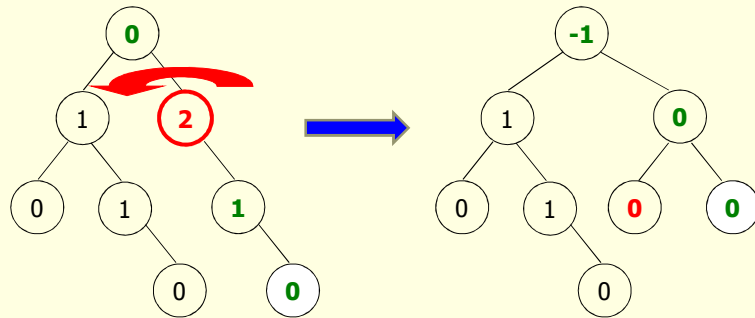
Questão

- Como cuidar disso?



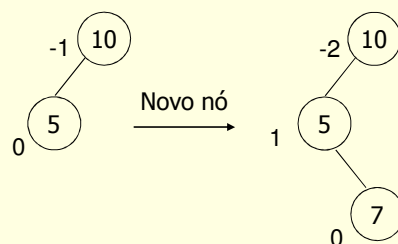
Questão

- Como cuidar disso?
 - Rotação simples para esquerda no local com problema
 - De forma ascendente, procura-se pelo primeiro 2/-2 a partir do local da inserção



21

AVL: exemplo de desbalanceamento

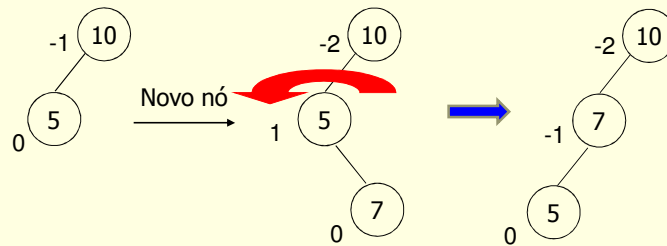


Desbalanceou!!!

Como arrumar isso?

22

AVL: exemplo de desbalanceamento

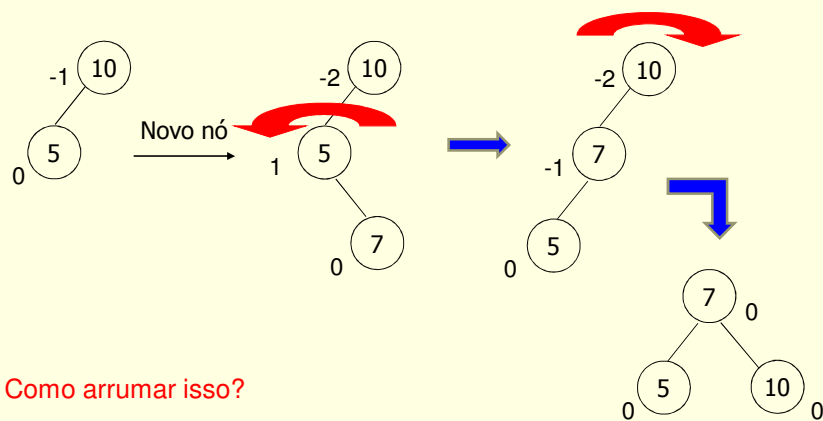


Como arrumar isso?

Rotação dupla: esquerda

23

AVL: exemplo de desbalanceamento

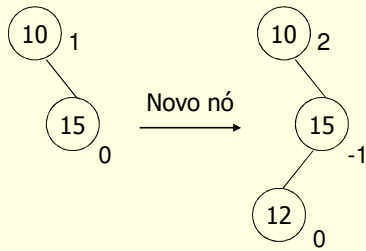


Como arrumar isso?

Rotação dupla: esquerda + direita

24

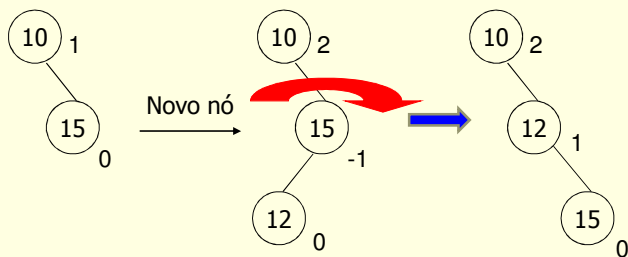
AVL: exemplo de desbalanceamento



Como arrumar isso?

25

AVL: exemplo de desbalanceamento

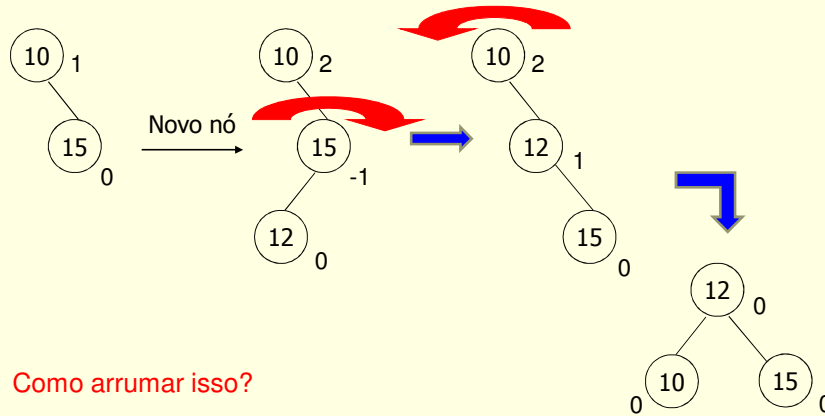


Como arrumar isso?

Rotação dupla: direita

26

AVL: exemplo de desbalanceamento

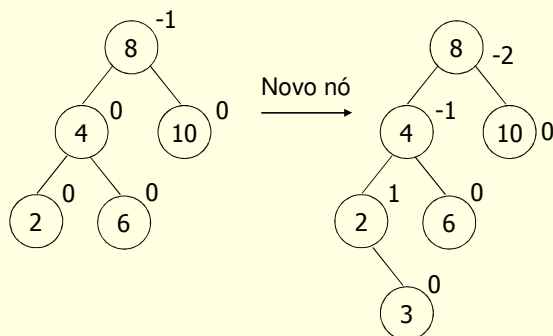


Como arrumar isso?

Rotação dupla: direita + esquerda

27

AVL: exemplo de desbalanceamento

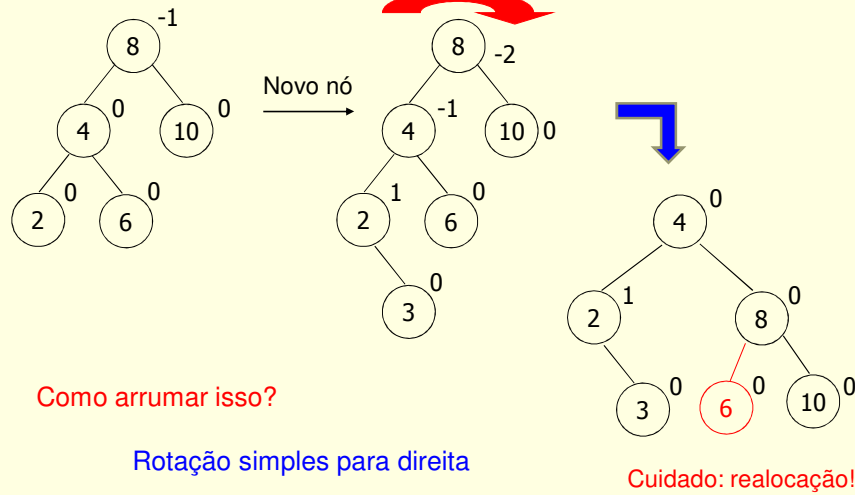


Como arrumar isso?

EXERCÍCIO

28

AVL: exemplo de desbalanceamento



AVL

■ Exercício

- Inserir os elementos 10, 3, 2, 5, 9, 7, 15, 12 e 13, nesta ordem, em uma árvore e balancear quando necessário

AVL

- Exercício
 - Inserir os elementos A, B, C, ..., J em uma árvore e balancear quando necessário

31

AVL

- Os **percursos** em-ordem da árvore original e da balanceada **permanecem iguais**
 - Exercício: prove para um dos exemplos anteriores!

32

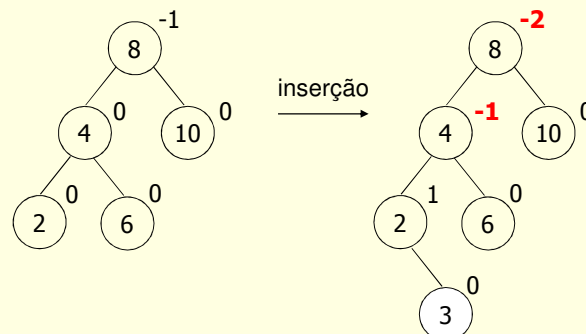
AVL

- Controle do balanceamento
 - Altera-se o algoritmo de inserção para balancear a árvore quando ela se tornar desbalanceada após uma inserção (nó com FB 2 ou -2)
 - Rotações
 - Se árvore pende para esquerda (FB negativo), rotaciona-se para a direita
 - Se árvore pende para direita (FB positivo), rotaciona-se para a esquerda
 - 2 casos podem acontecer

33

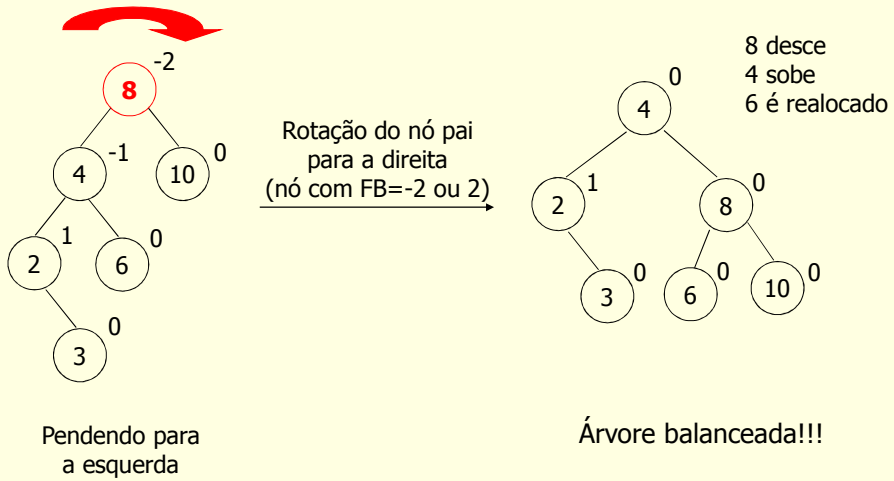
AVL: primeiro caso

- Raiz de uma subárvore com FB -2 (ou 2) e um nó filho com FB -1 (ou 1)
 - Os fatores de balanceamento têm **siniais iguais**: subárvores de nó raiz e filho pendem para o mesmo lado

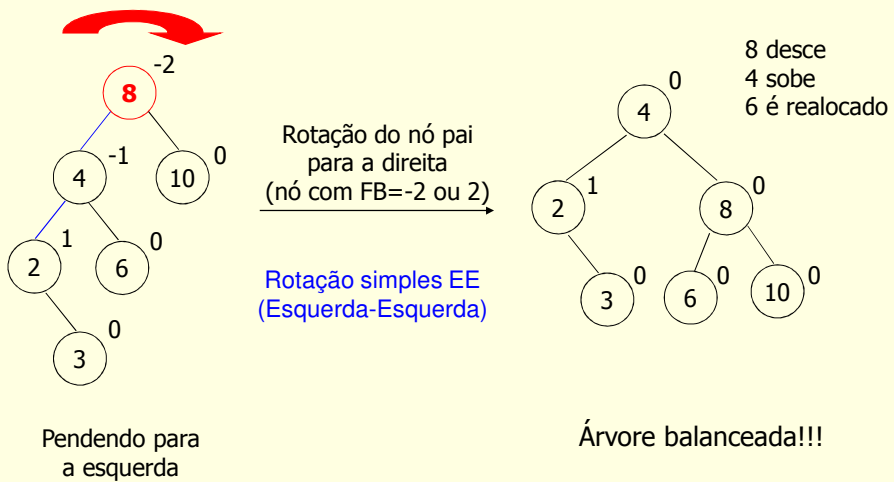


34

AVL: primeiro caso



AVL: primeiro caso



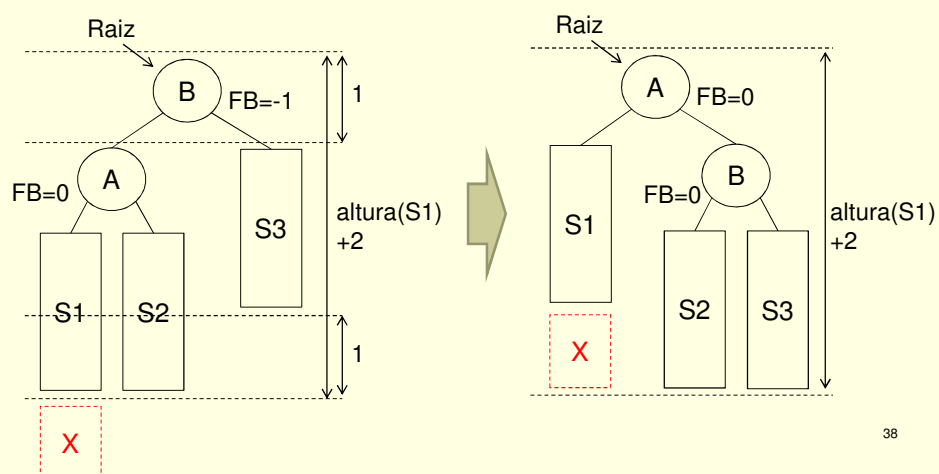
AVL: primeiro caso

- Quando subárvores do pai e filho pendem para um mesmo lado
 - Rotação simples para o lado oposto
 - EE ou DD (Direita-Direita, com raciocínio inverso)
- Às vezes, é necessário realocar algum elemento, pois ele perde seu lugar na árvore

37

Formalmente: EE

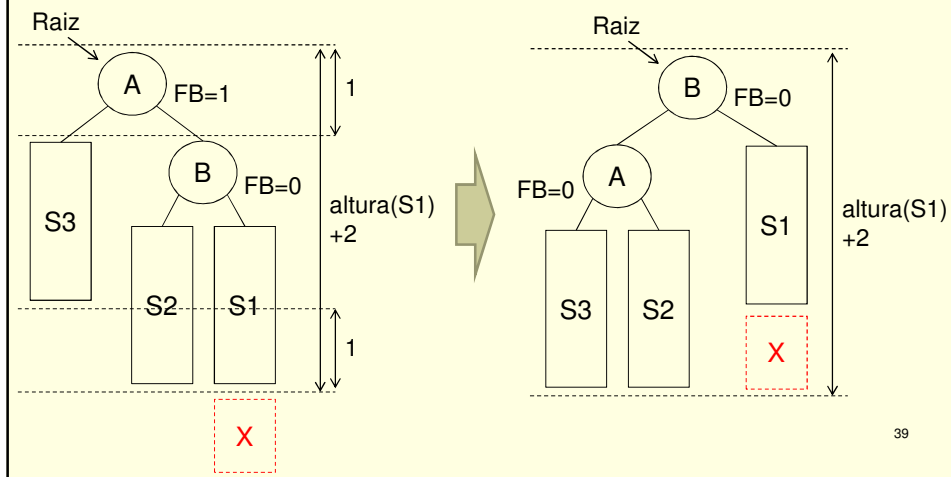
Generalizando



38

Formalmente: DD (similar)

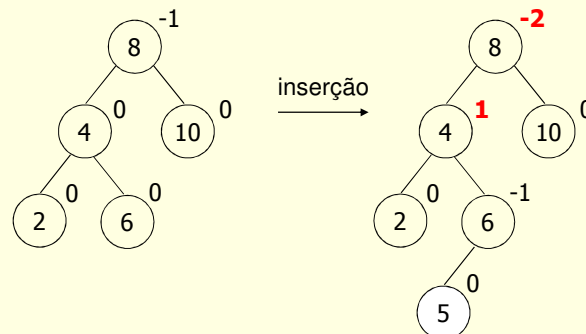
Generalizando



39

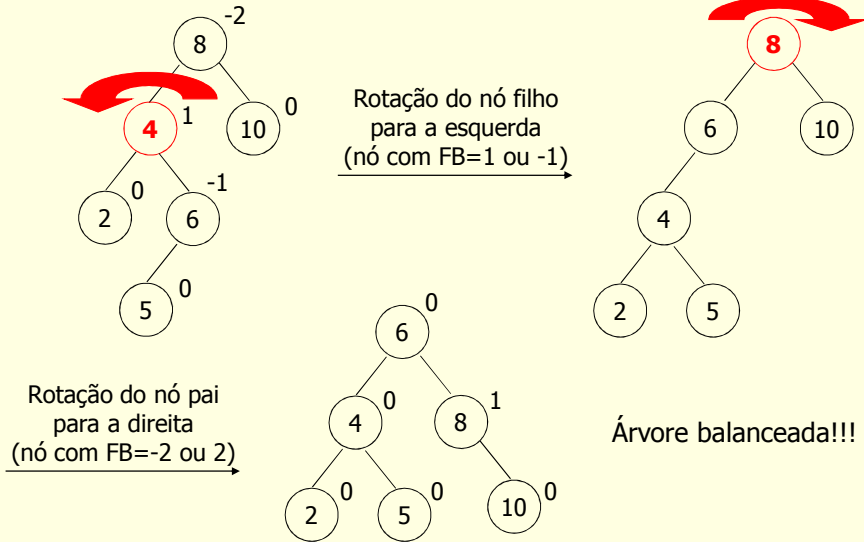
AVL: segundo caso

- Raiz de uma subárvore com FB -2 (ou 2) e um nó filho com FB 1 (ou -1)
 - Os fatores de balanceamento têm **siniais opostos**: subárvore de nó raiz pende para um lado e subárvore de nó filho pende para o outro (ou o contrário)

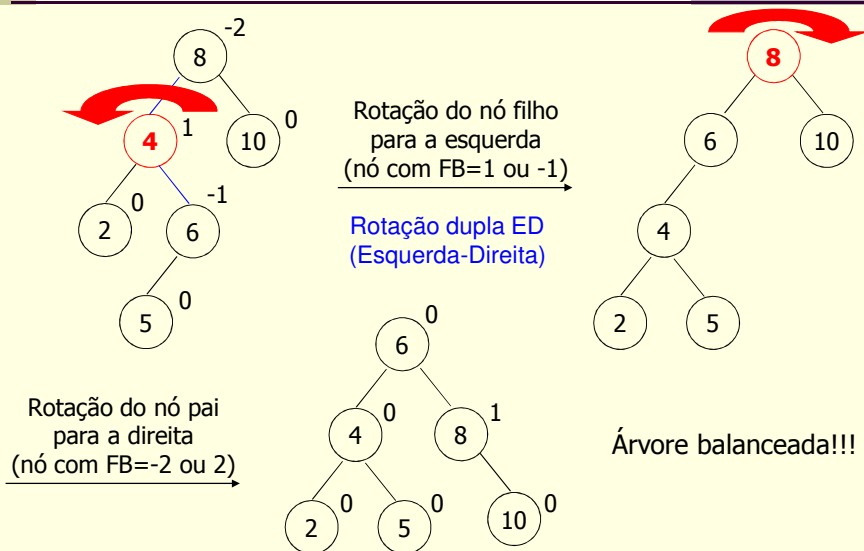


40

AVL: segundo caso



AVL: segundo caso

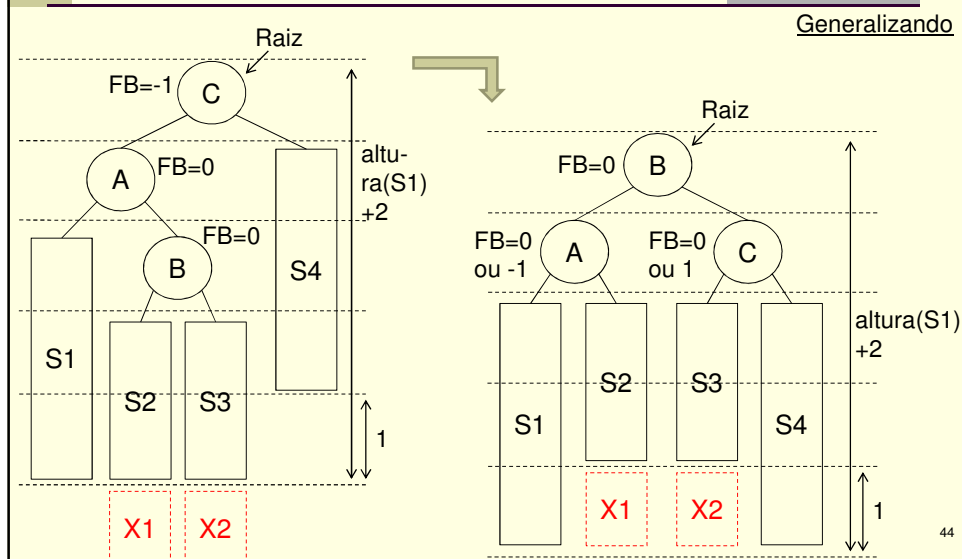


AVL: segundo caso

- Quando subárvores do pai e filho pendem para lados opostos
 - Rotação dupla
 - Primeiro, rotaciona-se o filho para o lado do desbalanceamento do pai
 - Em seguida, rotaciona-se o pai para o lado oposto do desbalanceamento
 - ED ou DE (com raciocínio inverso)
 - Às vezes, é necessário realocar algum elemento, pois ele perde seu lugar na árvore

43

Formalmente: ED



Formalmente: DE

- Exercício para casa

45

AVL

- As transformações dos casos anteriores diminuem em 1 a altura da subárvore com raiz desbalanceada p
- Assegura-se o rebalanceamento de todos os ancestrais de p e, portanto, o rebalanceamento da árvore toda

46

AVL

■ Novo algoritmo de inserção

- A cada inserção, verifica-se o balanceamento da árvore
 - Se necessário, fazem-se as rotações de acordo com o caso (sinais iguais ou não)
- Em geral, armazena-se uma variável de balanceamento em cada nó para indicar o FB

47

AVL

■ Declaração

```
typedef int elem;  
  
typedef struct bloco {  
    elem info;  
    struct bloco *esq, *dir;  
    int FB;  
} no;  
  
typedef struct {  
    no *raiz;  
} AVL;
```

48

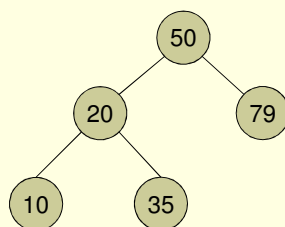
AVL

- Implementar sub-rotinas de inserção de elementos na AVL
 - Rotações
 - Função principal de inserção

49

AVL

- Exercício: teste a sub-rotina anterior inserindo os elementos **5** e **40** na árvore abaixo



50

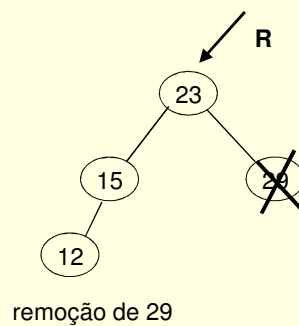
AVL

- **Exercício em duplas**
 - Pense no procedimento de remoção de um elemento da AVL e responda
 - Quais são os casos de desbalanceamento?
 - Esboce um método de remoção que balanceie a árvore se necessário

51

AVL: remoção

- Exemplos

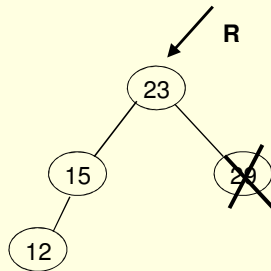


Como balancear?

52

AVL: remoção

■ Exemplos



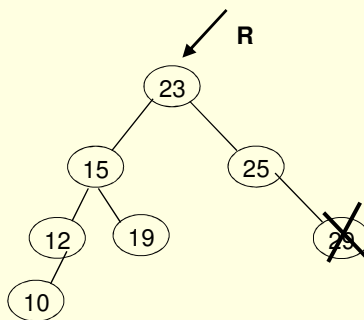
remoção de 29 = inserção de 12

Rotação simples EE

53

AVL: remoção

■ Exemplos



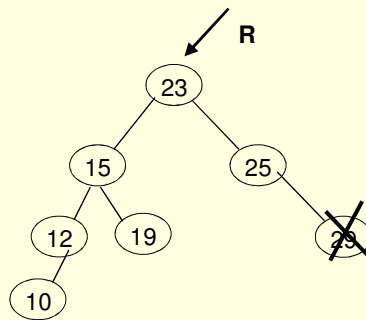
remoção de 29

Como balancear?

54

AVL: remoção

■ Exemplos



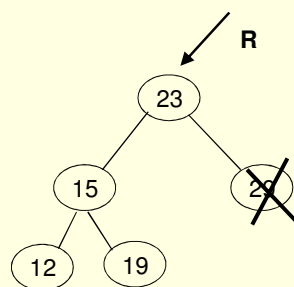
remoção de 29 = inserção de 10

Rotação simples EE

55

AVL: remoção

■ Exemplos



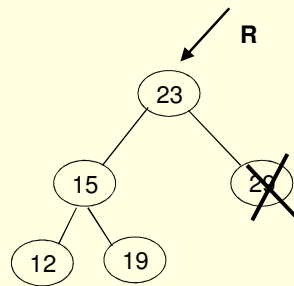
remoção de 29

Como balancear?

56

AVL: remoção

■ Exemplos



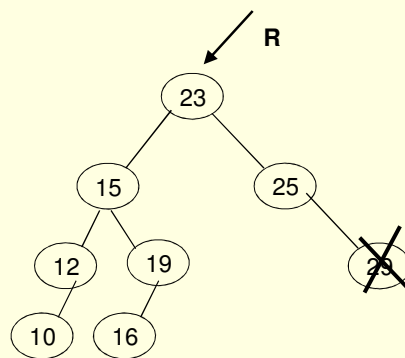
remoção de 29 = inserção de 12 ou 19
(não aconteceria na inserção)

Rotação simples EE

57

AVL: remoção

■ Exemplos



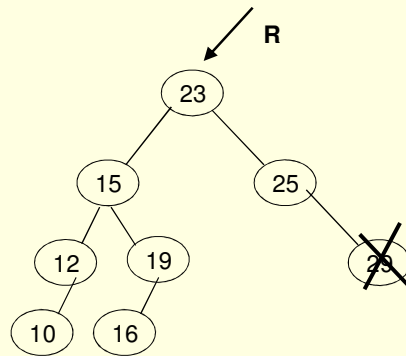
remoção de 29

Como balancear?

58

AVL: remoção

■ Exemplos



remoção de 29 = inserção de 10 ou 16
(não aconteceria na inserção)

Rotação simples EE

59

AVL: remoção

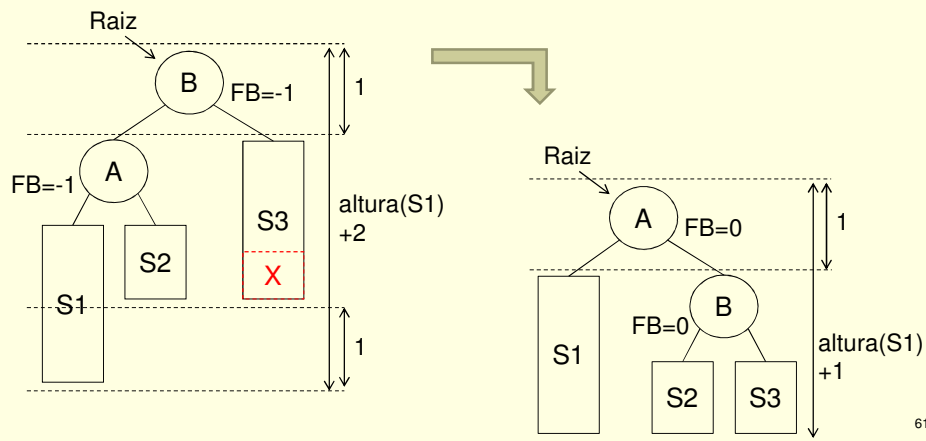
■ Primeiro caso

- Rotação simples em R (FB=2 ou -2) com filho com fator de balanceamento de mesmo sinal (1 ou -1) ou zero
 - Se R negativo, rotaciona-se para a direita; caso contrário, para a esquerda
- Rotação simples DD: raciocínio inverso

60

AVL: remoção

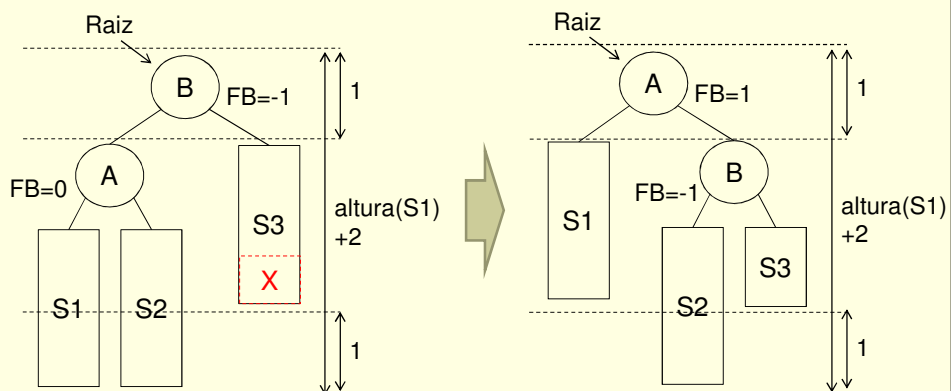
- Primeiro caso – situação 1: FB do filho = -1



61

AVL: remoção

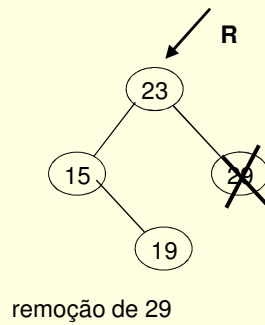
- Primeiro caso – situação 2: FB do filho=0



62

AVL: remoção

■ Exemplos

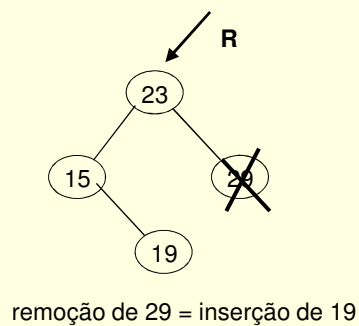


Como balancear?

63

AVL: remoção

■ Exemplos

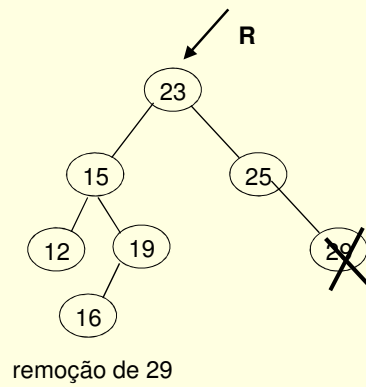


Rotação dupla ED

64

AVL: remoção

■ Exemplos

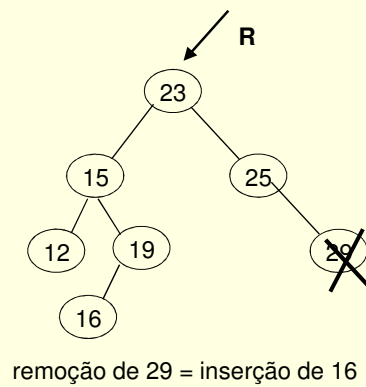


Como balancear?

65

AVL: remoção

■ Exemplos

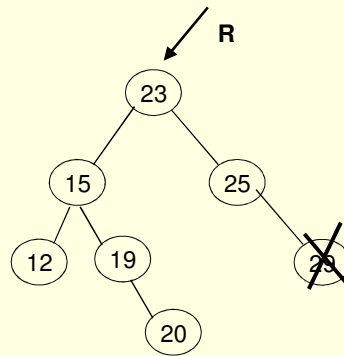


Rotação dupla ED

66

AVL: remoção

Exemplos



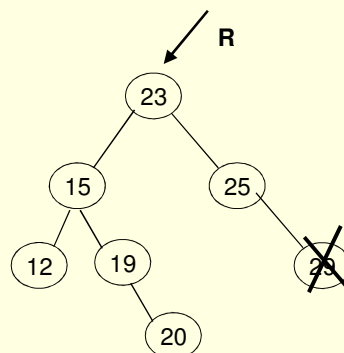
remoção de 29

Como balancear?

67

AVL: remoção

Exemplos



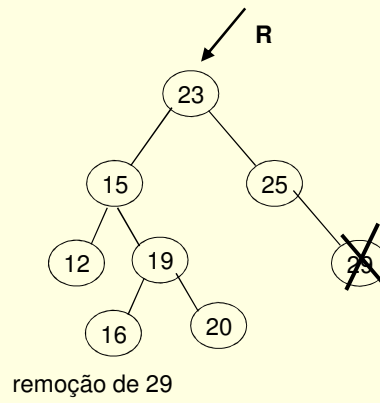
remoção de 29 = inserção de 20

Rotação dupla ED

68

AVL: remoção

■ Exemplos

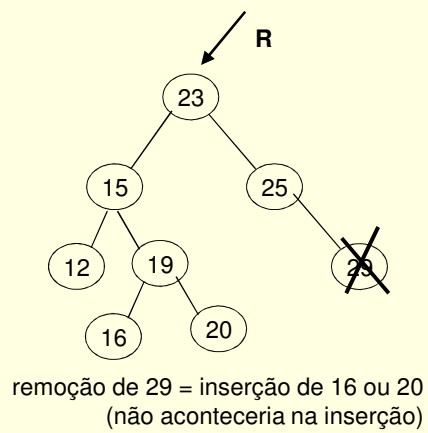


Como balancear?

69

AVL: remoção

■ Exemplos



Rotação dupla ED

70

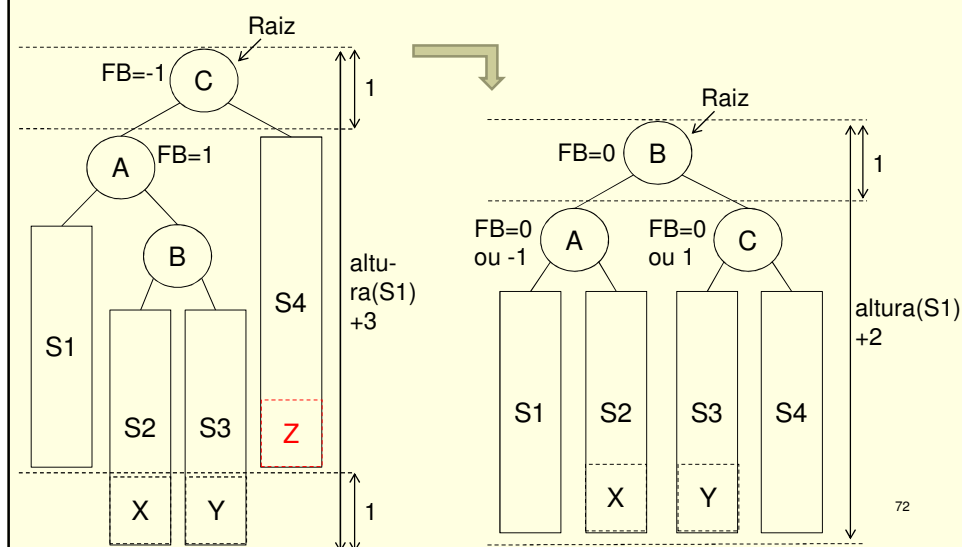
AVL: remoção

■ Segundo caso

- Rotação dupla quando R (FB=2 ou -2) e seu filho (1 ou -1) tem fatores de balanceamento com sinais opostos
 - Rotaciona-se o filho para o lado do desbalanceamento do pai
 - Rotaciona-se R para o lado oposto do desbalanceamento
- Rotação dupla DE: raciocínio inverso

71

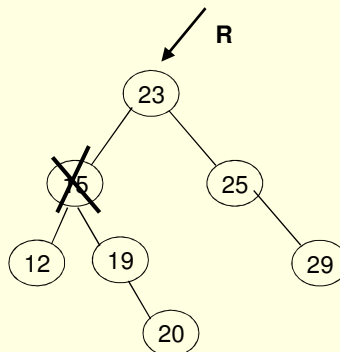
AVL: remoção



72

AVL: remoção

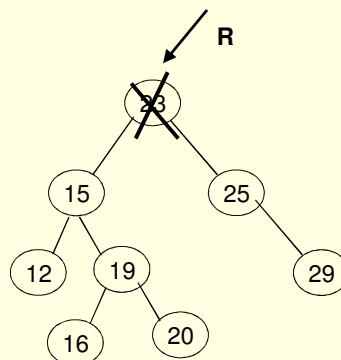
- Questão: como remover um nó intermediário em vez de um nó folha?
 - É necessário balancear?



73

AVL: remoção

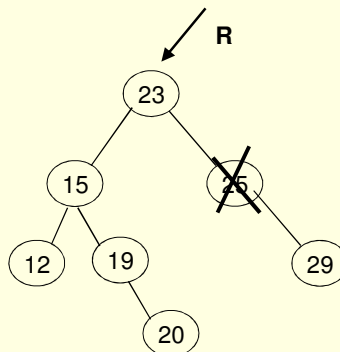
- Questão: como remover um nó intermediário em vez de um nó folha?
 - É necessário balancear?



74

AVL: remoção

- Questão: como remover um nó intermediário em vez de um nó folha?
 - É necessário balancear?



75

AVL: remoção

- **Exercício para casa**
 - Esquematizar cada caso de remoção em AVL
 - Que ponteiros mexer, como alterar os fatores de balanceamento
 - Esboçar sub-rotina de remoção de elemento de uma AVL

76