

# ESTRUTURA DE DADOS

## ARVORE BINARIA BALANCEADA

6/5/2008

Estutura de Dados  
Prof. Ademar Schmitz, M.Sc.

1

## BIBLIOGRAFIA

- GOODRICH, Michael. T., TAMASSIA, Roberto. **Estruturas de Dados e Algoritmos em Java**. São Paulo: Bookman, 2002.
  - Capítulo 6: Árvores.
- WEISS, Mark A. **Data Structures & Algorithm Analysis in Java**. Addison Wesley Longman, 1999.
  - Chapter 4: Trees

6/5/2008

Estutura de Dados  
Prof. Ademar Schmitz, M.Sc.

2

## INTRODUÇÃO

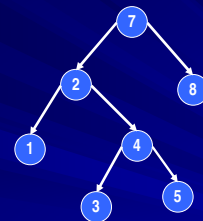
- Árvores balanceadas são também conhecidas como **AVL Trees**;
- AVL (Adelson-Velskii and Landis).
- Uma árvore AVL é idêntica a uma árvore binária, com a seguinte diferença:
  - Para cada nó na árvore, a diferença da altura da subárvore da direita e da esquerda deve ser de no máximo 1.

6/5/2008

Estutura de Dados  
Prof. Ademar Schmitz, M.Sc.

3

## INTRODUÇÃO



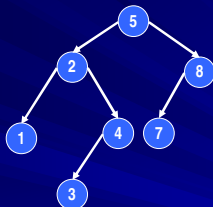
Esta árvore está balanceada?

6/5/2008

Estutura de Dados  
Prof. Ademar Schmitz, M.Sc.

4

## INTRODUÇÃO



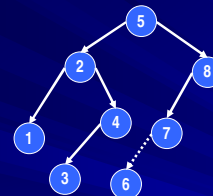
E esta?

6/5/2008

Estutura de Dados  
Prof. Ademar Schmitz, M.Sc.

5

## INTRODUÇÃO



E se quisermos inserir o número 6?

6/5/2008

Estutura de Dados  
Prof. Ademar Schmitz, M.Sc.

6

## ANÁLISE DAS OPERAÇÕES

- A **condição de balanceamento** garante que, com pequenas variações, todas as operações são realizadas em tempo  $O(\log n)$ .

6/5/2008

Estrutura de Dados  
Prof. Ademar Schmitz, M.Sc.

7

## ANÁLISE DAS OPERAÇÕES

- A inserção é mais demorada porque é na inserção que precisamos garantir a condição de balanceamento;
- Para a deleção podemos usar um método chamado **lazy deletion**;

6/5/2008

Estrutura de Dados  
Prof. Ademar Schmitz, M.Sc.

8

## CONDIÇÃO DE BALANCEAMENTO

- Depois de uma inserção, apenas nós que estão no caminho do ponto de inserção até a raiz da árvore podem ter sua condição de balanceamento alterada;

6/5/2008

Estrutura de Dados  
Prof. Ademar Schmitz, M.Sc.

9

## CONDIÇÃO DE BALANCEAMENTO

- Vamos chamar o nó que precisa ser rebalanceado de  $n$ ;
- Como um nó tem no máximo dois filhos, e a condição para **desbalanceamento** precisa que as duas subárvores de  $n$  tem uma diferença de 2, podemos identificar quatro formas de desbalancear uma árvore:

6/5/2008

Estrutura de Dados  
Prof. Ademar Schmitz, M.Sc.

10

## CONDIÇÃO DE BALANCEAMENTO

1. Uma inserção na subárvore da esquerda do filho da esquerda de  $n$  (**LEFT-LEFT**).
2. Uma inserção na subárvore da direita do filho da esquerda de  $n$  (**LEFT-RIGHT**).
3. Uma inserção na subárvore da esquerda do filho da direita de  $n$  (**RIGHT-LEFT**).
4. Uma inserção na subárvore da direita do filho da direita de  $n$  (**RIGHT-RIGHT**).

6/5/2008

Estrutura de Dados  
Prof. Ademar Schmitz, M.Sc.

11

## CONDIÇÃO DE BALANCEAMENTO

- Os casos 1 e 4 são simétricos em relação a  $n$  e assim o são os casos 2 e 3;
- Teoricamente temos apenas dois casos então;
- Na prática continuamos com 4 casos;

6/5/2008

Estrutura de Dados  
Prof. Ademar Schmitz, M.Sc.

12

## CONDIÇÃO DE BALANCEAMENTO

- O primeiro caso, no qual a inserção acontece do “lado de fora” (left-left, right-right), pode ser resolvido com **rotação simples**;
- O segundo caso, no qual a inserção acontece do “lado de dentro” (left-right, right-left), precisa ser resolvido com **rotação dupla**.

6/5/2008

Estutura de Dados  
Prof. Ademar Schmitz, M.Sc.

13

## CONDIÇÃO DE BALANCEAMENTO

- Caso 1: left-left
  - Rotação simples à direita.
- Caso 2: left-right
  - Rotação dupla esquerda-direita.
- Caso 3: right-left
  - Rotação dupla direita-esquerda.
- Caso 4: right-right
  - Rotação simples a esquerda.

6/5/2008

Estutura de Dados  
Prof. Ademar Schmitz, M.Sc.

14

## NOMENCLATURA

- $n$ : nodo desbalanceado.
- $n_l$ : filho da esquerda de  $n$ .
- $n_r$ : filho da direita de  $n$ .
- $L$ : árvore esquerda de  $n$ .
- $R$ : árvore direita de  $n$ .
- $LL$ : árvore esquerda de  $n_l$ .
- $LR$ : árvore direita de  $n_l$ .
- $RL$ : árvore esquerda de  $n_r$ .
- $RR$ : árvore direita de  $n_r$ .

6/5/2008

Estutura de Dados  
Prof. Ademar Schmitz, M.Sc.

15

## ROTAÇÃO SIMPLES

- Move uma subárvore para cima, e a outra para baixo;



A operação de rotação **simples à direita** segue os seguintes passos:

1.  $n_l$  é colocado na raiz.
2.  $LL$  permanece a subárvore esquerda de  $n_l$ .
3.  $n$  torna-se a raiz da subárvore direita de  $n_l$ .
4.  $LR$  é a nova subárvore esquerda de  $n$ .
5.  $R$  permanece a subárvore direita de  $n$ .

6/5/2008

Estutura de Dados  
Prof. Ademar Schmitz, M.Sc.

16

## ROTAÇÃO SIMPLES

- Move uma subárvore para cima, e a outra para baixo.



A operação de rotação **simples à esquerda** segue os seguintes passos:

1.  $n_r$  é colocado na raiz.
2.  $RR$  permanece a subárvore direita de  $n_r$ .
3.  $n$  torna-se a raiz da subárvore esquerda de  $n_r$ .
4.  $RL$  é a nova subárvore direita de  $n$ .
5.  $L$  permanece a subárvore esquerda de  $n$ .

6/5/2008

Estutura de Dados  
Prof. Ademar Schmitz, M.Sc.

17

## ROTAÇÃO SIMPLES

Insira estes números nesta sequência:  
3, 2, 1, 4, 5, 6, 7.

6/5/2008

Estutura de Dados  
Prof. Ademar Schmitz, M.Sc.

18

## ROTAÇÃO DUPLA



- A **rotação dupla esquerda-direita** é formada pelas seguintes etapas:
1. Rotação simples à esquerda com  $n_l$ .
  2. Rotação simples à direita com  $n$ .

6/5/2008

Estrutura de Dados  
Prof. Ademar Schmitz, M.Sc.

19

## ROTAÇÃO DUPLA



- A **rotação dupla direita-esquerda** é formada pelas seguintes etapas:
1. Rotação simples à direita com  $n_r$ .
  2. Rotação simples à esquerda com  $n$ .

6/5/2008

Estrutura de Dados  
Prof. Ademar Schmitz, M.Sc.

20

## ROTAÇÃO SIMPLES-DUPLA

Insira estes números nesta seqüência:

3, 2, 1, 4, 5, 6, 7.

16, 15, 14, 13, 12, 11, 10, 9, 8.

6/5/2008

Estrutura de Dados  
Prof. Ademar Schmitz, M.Sc.

21