

Aula 4

Estrutura de Dados

Prof. Vinicius Pozzobon Borin

1

Conversa Inicial

2

- Investigadores uma nova estrutura de dados
 - Árvore
 - Árvore binária
 - Árvore de Adelson-velskii e Landis (AVL)

3

Árvores Binárias

4

Uma busca eficiente

- Como implementar uma busca em um conjunto de dados gigantesco?
 - Array com busca binária? ☒
 - Lista encadeada? ☒

5

- Uma possível solução para esse tipo de problema é o emprego de uma estrutura não linear, a árvore, como uma árvore binária de busca

6

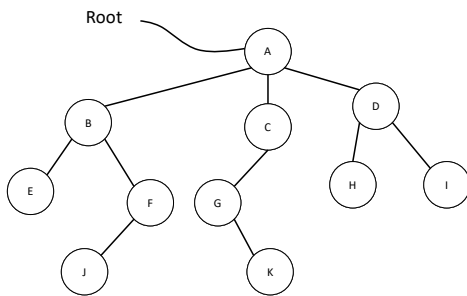
A árvore

- Uma estrutura de dados é denominada de **árvore** quando seus elementos criam ramificações na estrutura, gerando subárvores

7

- **Organização não linear**
- **Diferente de uma lista encadeada, que é linear**

8



9

Nomenclatura

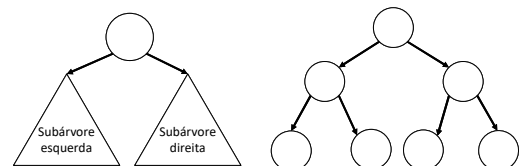
- **Nó raiz (*root*)** – nó original da árvore. Todos derivam dele
- **Nó pai** – nó que dá origem (está acima) a pelo menos um outro nó
- **Nó filho** – nó que deriva de um nó pai
- **Nó folha/terminal** – nó que não contém filhos

10

Árvore binária

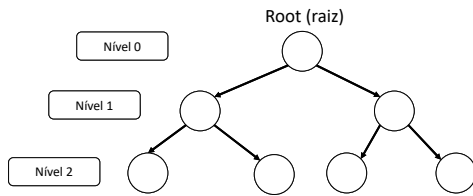
- Uma árvore é chamada de **binária** quando cada elemento (nó/nodo) da árvore contém, no máximo, dois (bi) filhos

11



12

Nível na árvore



13

Profundidade na árvore

- É a diferença entre o nível do nó e o nível da raiz

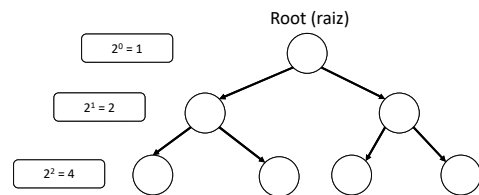
14

Altura na árvore

- É a diferença entre dois níveis quaisquer
- A altura é importante para balanceamento de árvore

15

Quantidade máxima de elementos na árvore



16

Nível	Nós no nível	Total de elemento por profundidade
0	$2^0 = 1$	$2^1 - 1 = 1$
1	$2^1 = 2$	$2^2 - 1 = 3$
2	$2^2 = 4$	$2^3 - 1 = 7$
3	$2^3 = 8$	$2^4 - 1 = 15$
4	$2^4 = 16$	$2^5 - 1 = 31$
5	$2^5 = 32$	$2^6 - 1 = 63$

$$Total_{\text{nós por profundidade}} = 2^{\text{nível}+1} - 1$$

17

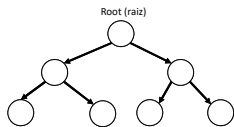
Tipos de árvores binárias

- Árvore estritamente binária
 - É aquela em que cada nó contém, sempre, exatamente dois filhos, ou nenhum
- Árvore binária completa
 - É aquela em que todos os nós contendo menos de dois filhos estão colocados somente no último ou no penúltimo nível da árvore

18

■ **Árvore binária cheia**

- É assim denominada quando cada nó tem exatamente dois filhos, e os nós folhas estão sempre no mesmo nível. Ou seja, é quando a árvore é estritamente binária e completa ao mesmo tempo



19

Árvore Binária de Busca

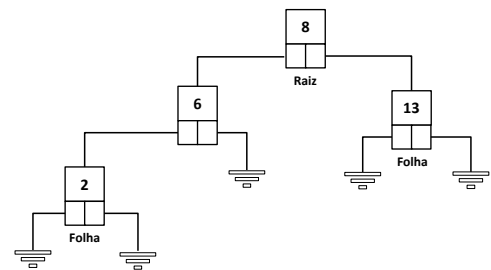
20

■ Também conhecida como

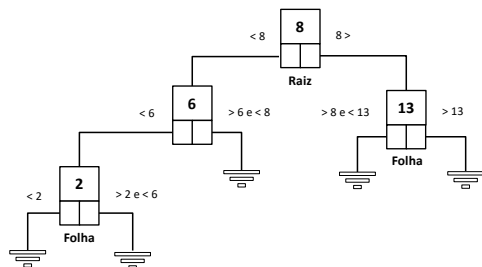
- **Binary Search Tree (BST)**

- Caso o valor a ser inserido seja menor do que o seu nó pai, inserimos o dado na subárvore esquerda
- Caso o valor a ser inserido seja maior do que o seu nó pai, inserimos o dado na subárvore direita

21

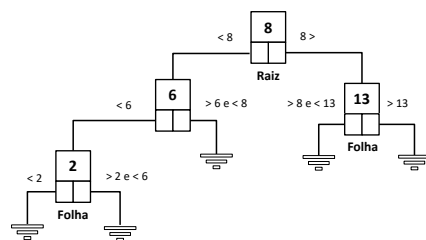


22

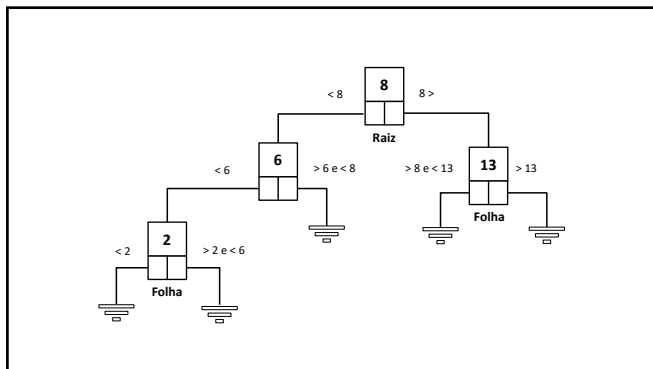


23

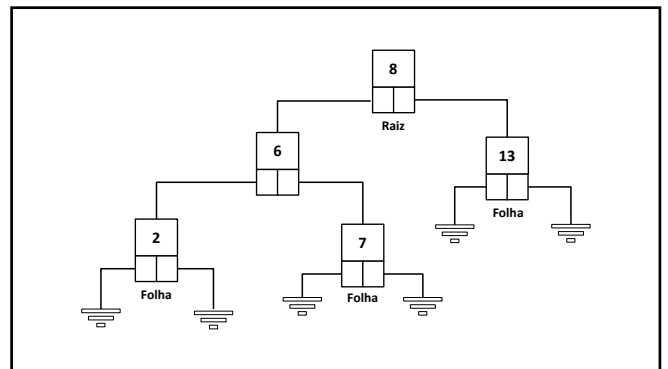
Inserindo na BST



24



25



26

▪ **Vejamos a implementação em Python**

27

Percurso em BST

28

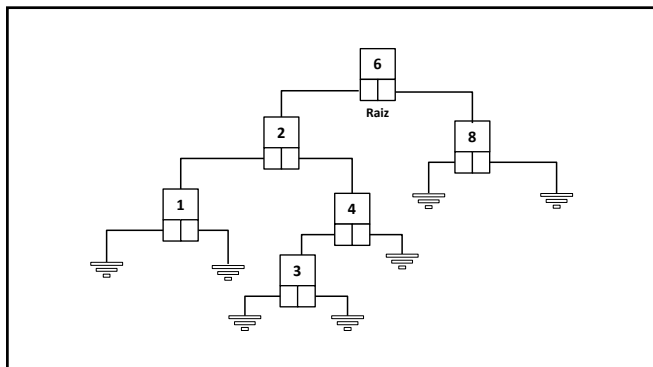
▪ **Podemos percorrer uma árvore de diferentes maneiras**

29

Percurso em largura

- **Level order**
- **Percorremos horizontalmente a árvore**
- **De cima para baixo, da esquerda para a direita**

30



31

- ▀ Resultado: 6, 2, 8, 1, 4, 3
- ▀ Vejamos a implementação em Python

32

Percurso em profundidade

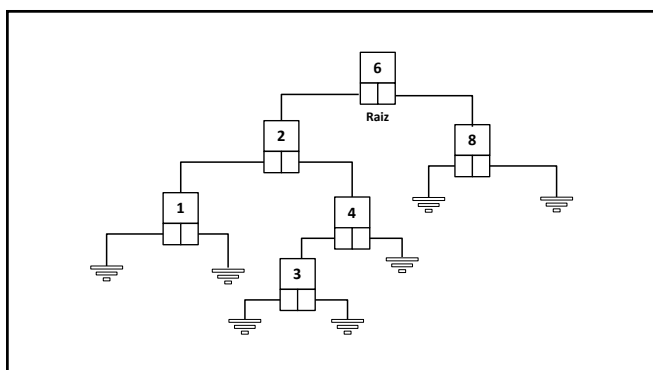
- ▀ É definido um caminho (galho da árvore), e vamos até o fim dele antes de retornarmos para escolher outro

33

Percurso em ordem

- ▀ Esquerda, raiz, direita

34



35

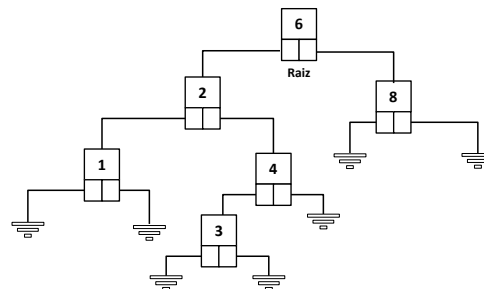
- ▀ Resultado: 1, 2, 3, 4, 6, 8
- ▀ Vejamos a implementação em Python

36

Percurso em pré-ordem

- Raiz, esquerda, direita

37



38

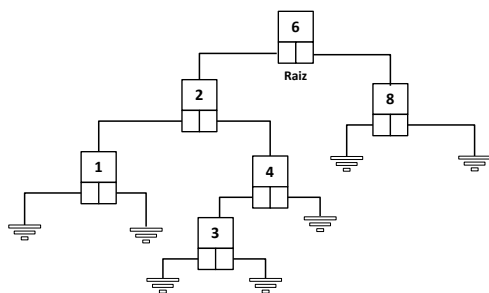
- Resultado: 6, 2, 1, 4, 3, 8
- Vejamos a implementação em Python

39

Percurso em pós-ordem

- Esquerda, direita, raiz

40



41

- Resultado: 1, 3, 4, 2, 8, 6
- Vejamos a implementação em Python

42

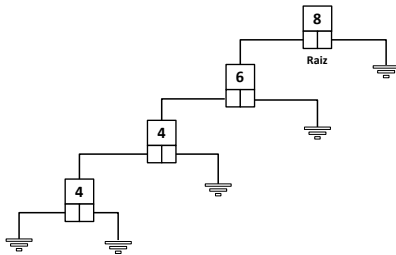
Árvore AVL

43

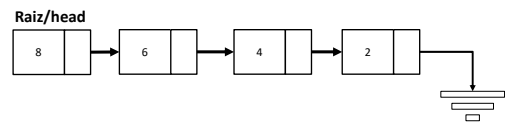
- ▀ Também conhecida como
 - Árvore AVL
 - Árvore binária de busca balanceada
- ▀ O que é uma árvore balanceada?

44

BST desbalanceada



45



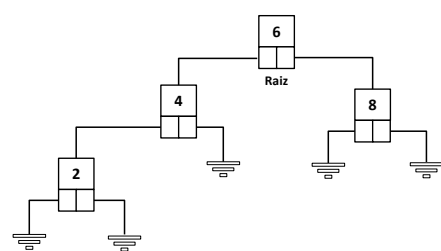
46

Árvore AVL

- ▀ Tem como objetivo melhorar o desempenho de busca/varredura, balanceando uma árvore binária, evitando ramos longos e gerando o maior número de ramificações binárias possíveis

47

BST balanceada



48

Características da árvore AVL

- Mesmas características da BST
- Uma propriedade a mais
 - A diferença de altura entre a subárvore da direita e da subárvore da esquerda será sempre 0, 1 ou -1

49

O balanceamento

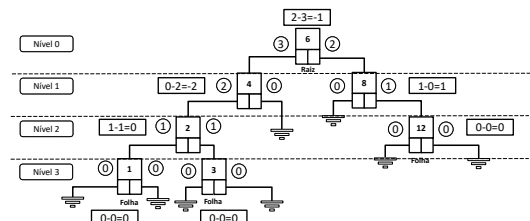
- Passo 1: calcula-se a altura relativa daquele elemento para o lado direito da árvore. Nesse caso, pegamos o nível mais alto do lado direito daquele elemento e subtraímos do nível do elemento desejado
- Passo 2: calcula-se a altura relativa daquele elemento para o lado esquerdo da árvore. Nesse caso, pegamos o nível mais alto do lado esquerdo daquele elemento e subtraímos do nível do elemento desejado

50

- Passo 3: tendo a altura direita e a esquerda calculada, fazemos a diferença entre elas (direita menos esquerda, sempre). Se o cálculo resultar em 2 ou -2, existe um desbalanceamento e uma rotação será necessária na árvore

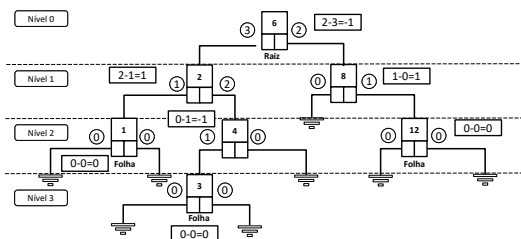
51

Exemplo desbalanceado



52

Exemplo balanceado



53

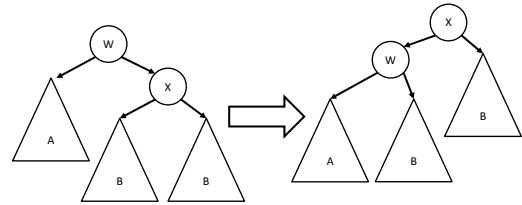
Rotações da Árvore AVL

54

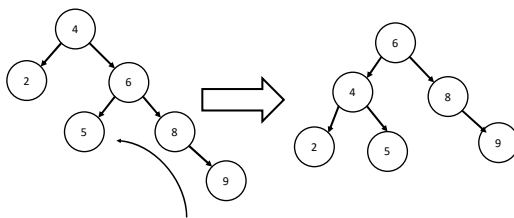
Diferença de altura de um nó	Diferença de altura do nó filho do nó desbalanceado	Tipo de rotação
2	1	Simples à esquerda
2	0	Simples à esquerda
2	-1	Dupla com filho para a direita e pai para a esquerda
-2	1	Dupla com filho para a esquerda e pai para a direita
-2	0	Simples à direita
-2	-1	Simples à direita

55

Rotação simples à esquerda

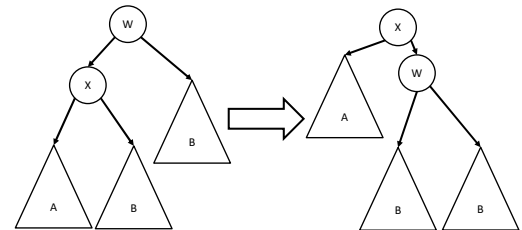


56

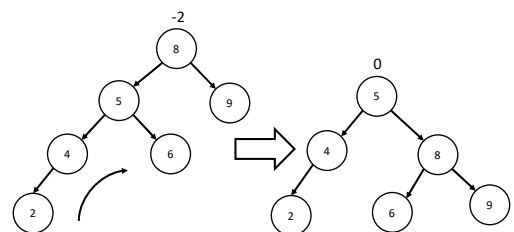


57

Rotação simples à direita



58



59

Complexidade Big-O

	Acesso ao dado	Busca	Inserção	Remoção
Árvore Binária	$O(n)$	$O(n)$	$O(n)$	$O(n)$
BST	$O(n)$	$O(n)$	$O(n)$	$O(n)$
AVL	$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(\log n)$

60

Referências

- ASCENCIO, A. F. G.; ARAÚJO, G. S. Estruturas de dados: algoritmos, análise da complexidade e implementações em JAVA e C/C++. São Paulo: Pearson Prentice, 2010.
- BHARGAVA, A. Y. Entendendo algoritmos. São Paulo: Novatec, 2017.

61

62

- DROZDEK, A. Estrutura de dados e algoritmos em C++. Tradução da 4ª edição norte-americana. São Paulo: Cengage Learning Brasil, 2018.
- FERRARI, R. *et al.* Estruturas de dados com jogos. Rio de Janeiro: Elsevier, 2014.
- KOFFMAN, E. B.; WOLFGANG, P. A. T. Objetos, abstração, estrutura de dados e projeto usando C++. Barueri: Grupo GEN, 2008.

63

64