

Técnicas de programação para Games

Aula09

Árvores

Professor Mestre: Adilson Lopes Khouri

2 de novembro de 2018

Sumário

Árvores

Cronograma

Aula	Conteúdo
12/04/2018	XP e banco de dados
17/04/2018	Introdução de estruturas de dados
24/04/2018	Arrays / Matrizes e Ordenação
26/04/2018	Recursão
03/05/2018	Lista Ligada
08/05/2018	Pilha, Fila
10/05/2018	Hash
15/05/2018	Árvore Binária
17/05/2018	Heap
22/05/2018	Grafos
24/05/2018	Prova

Árvores Binárias de Busca

- ▶ São estruturas de dados que permitem inserções, deleções e busca em complexidade logarítmica.
- ▶ $O(\log n)$ para inserir, atualizar, deletar e buscar
- ▶ É uma estrutura recursiva onde cada Nó contém outros dois nós, denominados filhos.

Árvores Binárias de Busca

- ▶ Esse tipo de estrutura de dados segue uma regra simples, os filhos da esquerda devem ser menores que seu pai. Os filhos da direita devem ser maiores que seu pai.
- ▶ Em outras palavras, ao inserir um novo nó devemos procurar a posição correta desse na árvore

Árvores Binárias de Busca

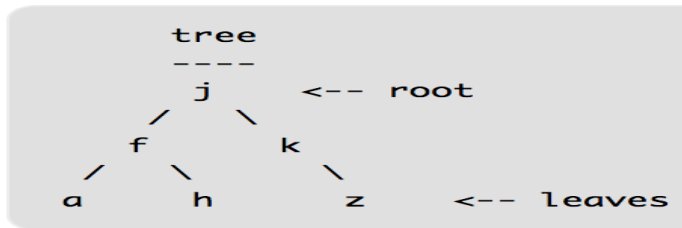


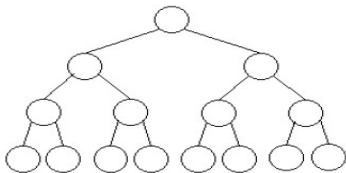
Figura: Exemplo de árvore binária de busca

Árvores Binárias de Busca

- ▶ Algumas propriedades das árvores binárias:
- ▶ O número máximo de nós no nível i é dado por: 2^{i-1} Nível aqui é o número de nós entre a raiz e o nó em questão
- ▶ A altura da árvore é $\log_2 n$
- ▶ Uma árvore binária de altura h tem no máximo $2^h - 1$ nós

Árvores Binárias de Busca

- Exemplo de árvore balanceada completa:



Exemplo de árvore cheia – possuindo o número máximo de nós para a sua altura.

Figura: Exemplo de árvore balanceada

Árvores Binárias de Busca

- ▶ Exemplo de árvore desbalanceada:

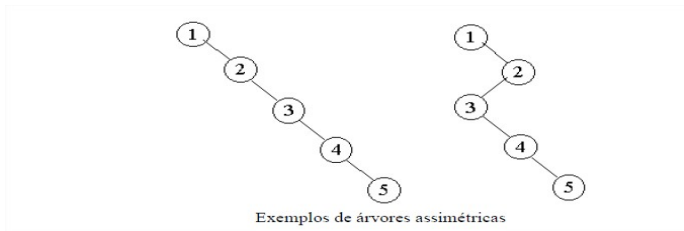


Figura: Exemplo de árvore desbalanceada

Árvores Binárias de Busca

- ▶ Para percorrer árvores binárias de busca, temos três opções de percurso:
- ▶ Em ordem
- ▶ Pré-ordem
- ▶ Pós-ordem

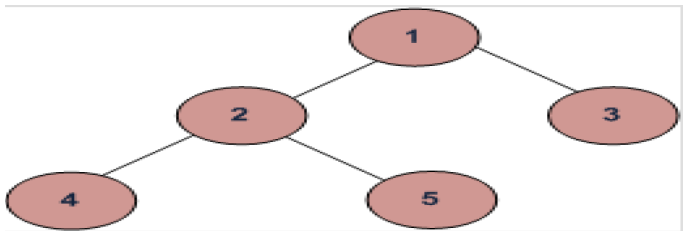


Figura: Exemplo de árvore

Imprime em Ordem

- ▶ Exemplo de árvore para percorrermos:
- ▶ Inorder (Left, Root, Right) : 4 2 5 1 3

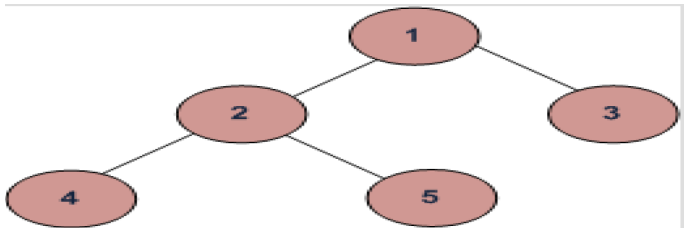


Figura: Exemplo de árvore

Imprimir árvore em ordem:

```
1  /* Given a binary tree, print its nodes in inorder*/
2      void imprimeEmOrdem(Node node)
3      {
4          if (node == null)
5              return;
6
7          /* first recur on left child */
8          imprimeEmOrdem(node.left);
9
10         /* then print the data of node */
11         System.out.print(node.key + " ");
12
13         /* now recur on right child */
14         imprimeEmOrdem(node.right);
15     }
```

Imprime Pós-Ordem

- ▶ Exemplo de árvore para percorrermos:
- ▶ Inorder (Left, Root, Right) : 4 5 2 3 1

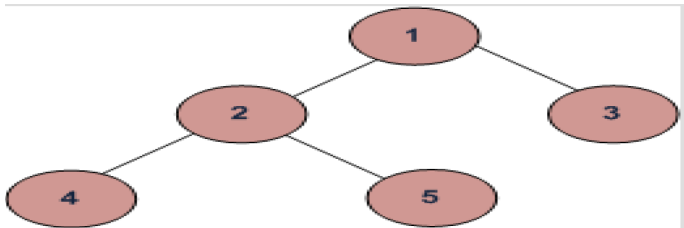


Figura: Exemplo de árvore

Imprimir árvore pós-ordem:

```
1      /* Given a binary tree, print its nodes according to the  
2 "bottom-up" postorder traversal. */  
3      void imprimePosOrdem(Node node)  
4      {  
5          if (node == null)  
6              return;  
7  
8          // first recur on left subtree  
9          imprimePosOrdem(node.left);  
10  
11         // then recur on right subtree  
12         imprimePosOrdem(node.right);  
13  
14         // now deal with the node  
15         System.out.print(node.key + " ");  
16     }  
17
```

Imprime Pré-Ordem

- ▶ Exemplo de árvore para percorrermos:
- ▶ Inorder (Left, Root, Right) : 1 2 4 5 3

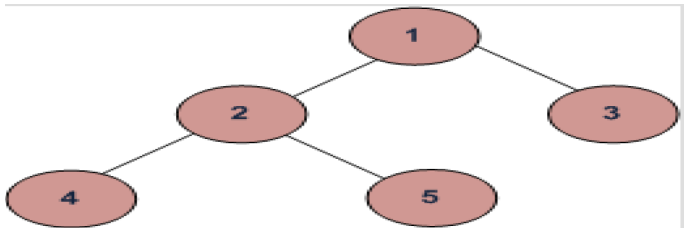


Figura: Exemplo de árvore

Imprimir árvore pré-ordem:

```
1  /* Given a binary tree, print its nodes in preorder*/
2      void imprimePreOrdem(Node node)
3      {
4          if (node == null)
5              return;
6
7          /* first print data of node */
8          System.out.print(node.key + " ");
9
10         /* then recur on left subtree */
11         imprimePreOrdem(node.left);
12
13         /* now recur on right subtree */
14         imprimePreOrdem(node.right);
15     }
```


Árvores Binárias de Busca

- ▶ Para pesquisar em árvores:
- ▶ Compare a chave pesquisada com a raiz, se for menor vá para a direita, caso contrário vá para a esquerda.
- ▶ Repita o processo novamente até encontrar o nó pesquisado ou chegar em uma folha e não encontrar o valor.

Imprime Pré-Ordem

- ▶ Simular no quadro a pesquisa pela chave: "7":

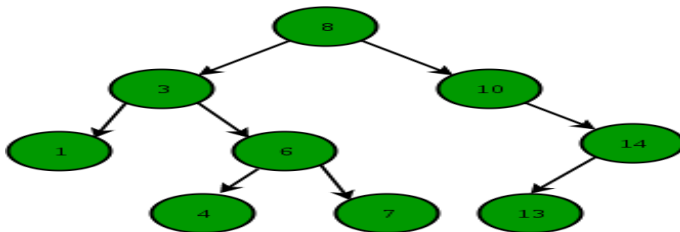


Figura: Exemplo de árvore

Imprime Pré-Ordem

- ▶ Como inserir valores?
- ▶ Percorra a árvore até encontrar a folha que será pai do novo nó.
- ▶ Insira uma nova folha na árvore e acerte os ponteiros.

Imprime Pré-Ordem

- Exemplo de inserção do valor: "40" na árvore:

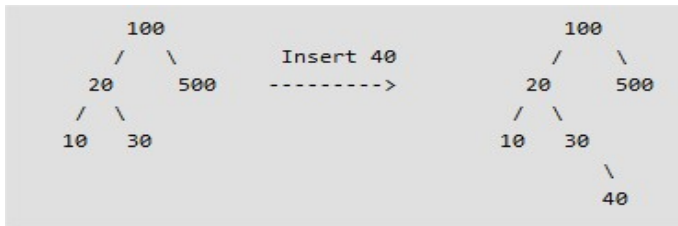


Figura: Exemplo de árvore

Imprime Pré-Ordem

- ▶ Como podemos deletar nós? Temos três casos de deleção:
- ▶ Deletar uma folha
- ▶ Nó para ser deletado tem apenas um filho
- ▶ Nó para ser deletado tem dois filhos

Imprime Pré-Ordem

- Deletar uma folha, simplesmente delete (caso simples)

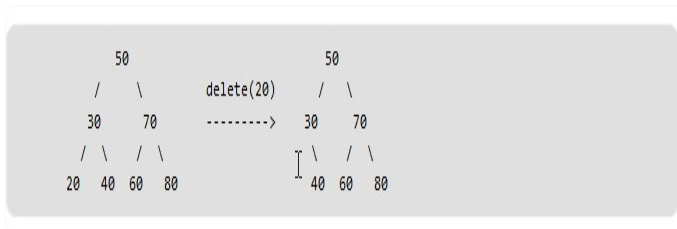


Figura: Exemplo de árvore

Imprime Pré-Ordem

- ▶ Deletar nó com apenas um filho, troque o valor do filho com o pai, e delete o filho (com valor trocado).

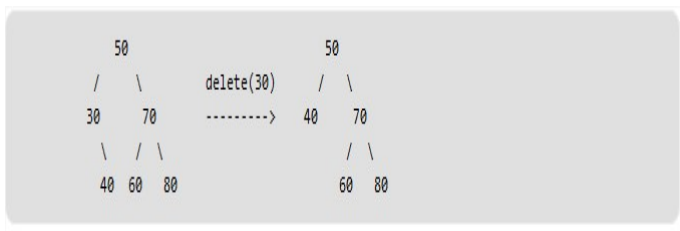


Figura: Exemplo de árvore

Imprime Pré-Ordem

- ▶ Deletar nó com dois filhos, encontre o sucessor em ordem, troque seus valores com o pai e delete o sucessor (com valor trocado)

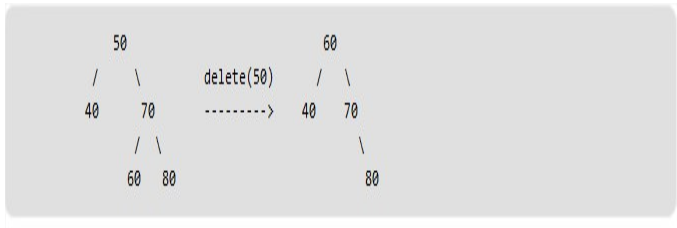


Figura: Exemplo de árvore

Exemplo de deleção

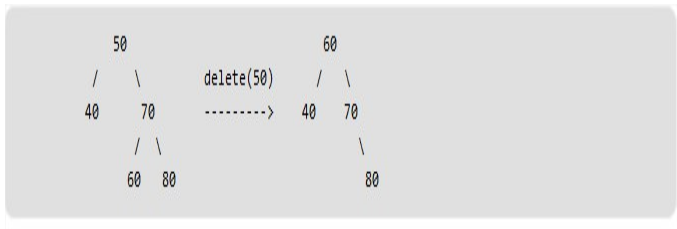


Figura: Exemplo de árvore

Exemplo de deleção

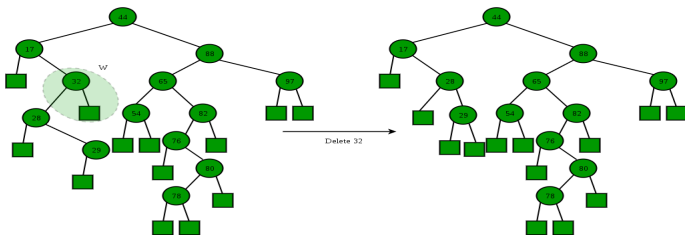


Figura: Exemplo de árvore

Exemplo de deleção

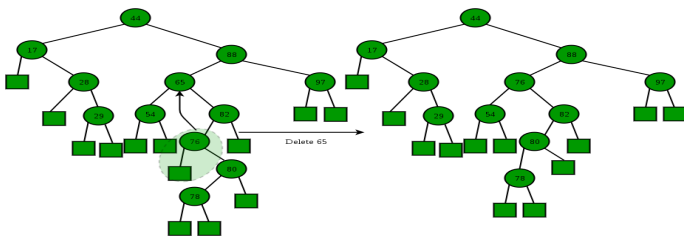


Figura: Exemplo de árvore

AVL

- ▶ Percebemos que manter a árvore balanceada é um desafio. Se a árvore desbalancear podemos cair no caso de uma busca sequencial em lista ligada.
- ▶ Como garantir que a árvore não vai ficar desbalanceada?
- ▶ Para garantir isso existe a estrutura denominada AVL, é uma árvore binária de busca com as mesmas regras que já estudamos. Porém há uma vantagem, ela se auto balanceia a cada inserção em tempo $\log n$

AVL

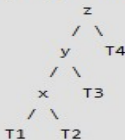
- ▶ A diferença principal aqui é que sempre que inserimos ou deletamos precisamos avaliar se ocorreu um desbalanceamento. Se ocorreu, precisamos rebalancear a árvore.
- ▶ Existem quatro casos de rotação que podemos aplicar, dependendo do caso de inserção.

Rotação: Left Left

► Left Left

a) Left Left Case

T1, T2, T3 and T4 are subtrees.



Right Rotate (z)

----->

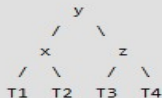


Figura: Exemplo de rotação left - left

Rotação: Left Righth

► Left Righth

b) Left Right Case

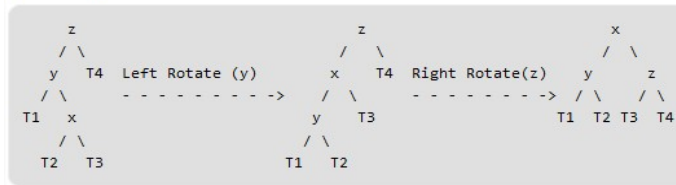


Figura: Exemplo de rotação left - right

Rotação: Righth Righth

► Righth Righth

c) Right Right Case

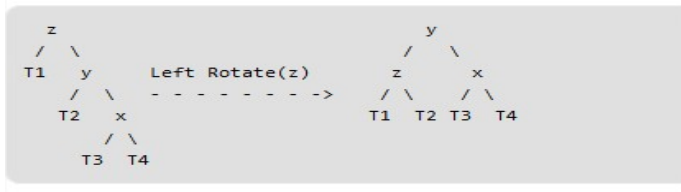


Figura: Exemplo de rotação right - righth

Rotação: Righth Left

► Righth Left

d) Right Left Case

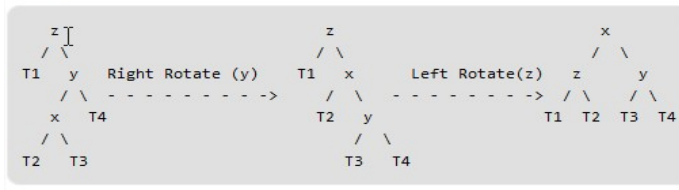


Figura: Exemplo de rotação right - left

AVL

- ▶ Como determinar o caso de inserção de uma árvore com raiz "z" novo nó "w"?
- ▶ Simples:
- ▶ y é o filho da esquerda de z e x é o filho da esquerda de y(LL)
- ▶ y é o filho da esquerda de z e x é o filho da direita de y(LR)
- ▶ y é o filho da direita de z e x é o filho da direita de y(RR)
- ▶ y é o filho da direita de z e x é o filho da esquerda de y(RL)

Rotação: Right Left

d) Right Left Case

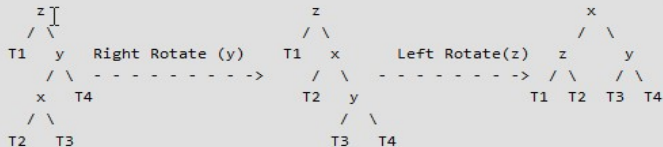


Figura: Exemplo de rotação right - left

Exemplo de inserção

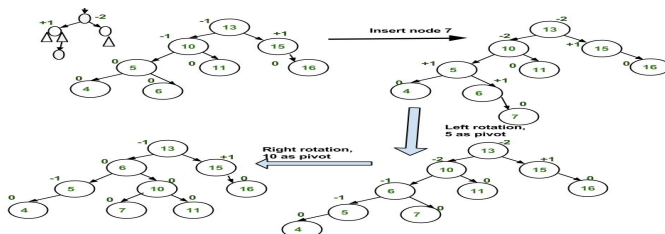


Figura: Exemplo de inserção

Exemplo de inserção

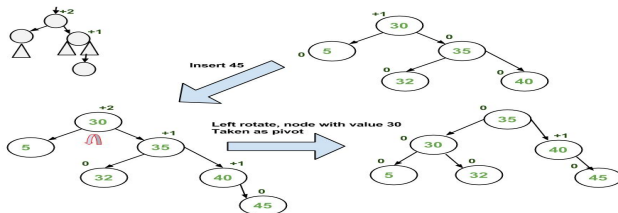


Figura: Exemplo de inserção

Exemplo de inserção

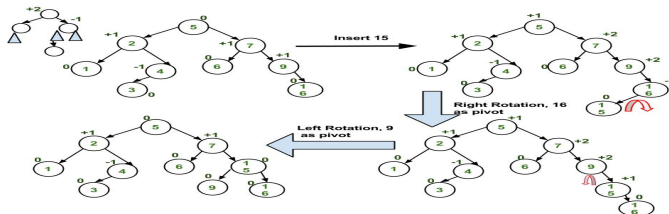


Figura: Exemplo de inserção

Exemplo de inserção

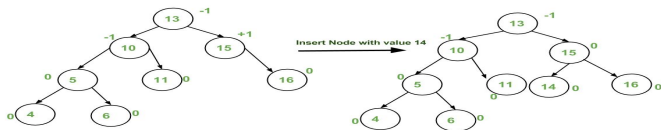


Figura: Exemplo de inserção

Exemplo de inserção

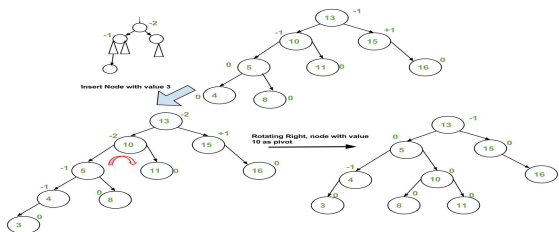


Figura: Exemplo de inserção

Dúvidas...

Alguma dúvida?

Contato

- ▶ E-mail: *0800dirso@gmail.com* (alunos SENAC)
- ▶ E-mail: *adilson.khoury.usp@gmail.com*
- ▶ Phone: +55119444 – 26191
- ▶ [Linkedin](#)
- ▶ [Lattes](#)
- ▶ [GitHub](#)

- [1] A. V. Aho, J. E. Hopcroft, and J. Ullman, *Data Structures and Algorithms*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1983.
- [2] K. Beck and C. Andres, *Extreme Programming Explained: Embrace Change (2Nd Edition)*. Addison-Wesley Professional, 2004.
- [3] Beck, *Test Driven Development: By Example*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002.
- [4] M. Fowler, *Refactoring: Improving the Design of Existing Code*. Boston, MA, USA: Addison-Wesley, 1999.
- [5] Geeks. (2018) A computer science portal for geeks. [Online]. Available: <https://www.geeksforgeeks.org>
- [6] Fernanda. (2014) Sql join: Entenda como funciona o retorno dos dados. [Online]. Available: <https://www.devmedia.com.br/sql-join-entenda-como-funciona-o-retorno-dos-dados/31006>
- [7] S. team. (2018) Sqlite sample database. [Online]. Available: <http://www.sqlitetutorial.net/sqlite-sample-database/>