

SIN 211 - Algoritmos e Estruturas de Dados

(Árvores Balanceadas)

Prof^o: Joelson Antônio dos Santos

Universidade Federal de Viçosa
Instituto de Ciências Exatas e Tecnológicas
Campus de Rio Paranaíba - MG

joelsonn.santos@gmail.com
Sala: BBT 233

19 de junho de 2018

Aula de Hoje

1 Árvores Balanceadas

- Árvores n -árias
- Árvore AVL

Árvores n -árias

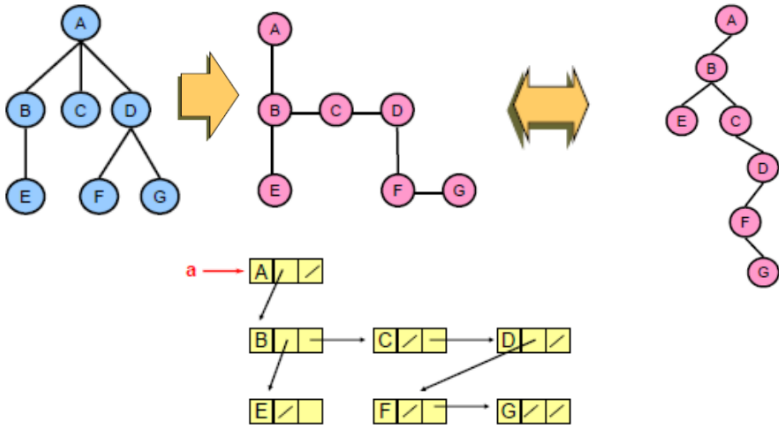
- Em uma árvore n -ária, cada nó pode ter um número diferente de subárvores;
- Uma forma de implementação seria definir um número máximo fixo para o grau da árvore;
 - **Desvantagem:** Alguns ponteiros poderiam nunca ser utilizados; Qual a solução para isso?

Árvores n -árias

- Uma forma de solucionar esse problema é transformando uma árvore n -ária em uma árvore binária;
- Para isso os passos seguintes devem ser considerados:
 - O primeiro filho de um nó passa a ser o filho à esquerda na árvore binária.
 - Os demais filhos de um nó passam a ser filhos à direita do seu irmão imediato à esquerda.
 - O processo é repetido para cada nó da árvore.

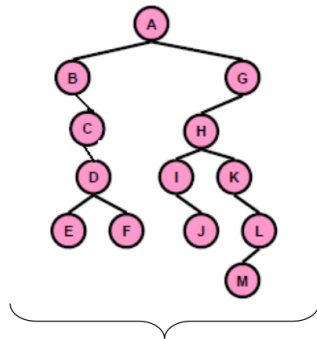
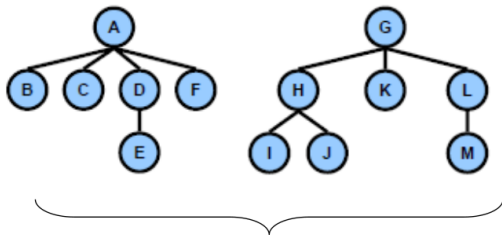
Árvores n-árias

- Conversão para árvore binária;



Árvores n-árias

- Para converter uma floresta basta considerar as raízes como nós irmãos;



Relembrando: Árvores Binárias de Pesquisa

- Uma árvore binária de pesquisa é sensível à ordem de inserção dos elementos;
- Diferentes árvores podem ser construídas dependendo da ordem de inserção, o que pode tornar a árvore eficiente ou não.
- **Exemplo:** Se fossem inseridos o conjunto de elementos $\{1, 2, 3, 4, 5, 6, 7\}$, como ficaria a árvore binária de pesquisa?

Relembrando: Árvores Binárias de Pesquisa

- E se fossem inseridos os elementos na ordem seguinte?

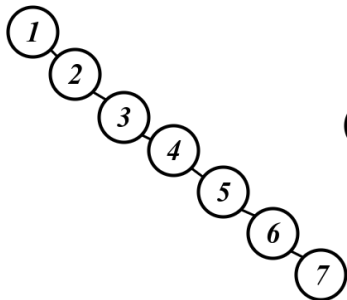
Relembrando: Árvores Binárias de Pesquisa

- E se fossem inseridos os elementos na ordem seguinte?
 - {4, 2, 6, 1, 3, 5, 7}

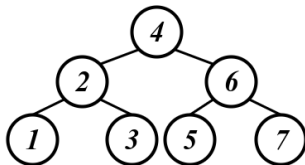
Relembrando: Árvores Binárias de Pesquisa

- E se fossem inseridos os elementos na ordem seguinte?
 - {4, 2, 6, 1, 3, 5, 7}

inserção: 1,2,3,4,5,6,7



inserção: 4,2,6,1,3,5,7



Árvores AVL

- A ideia é manter uma árvore sempre balanceada para qualquer nó da árvore;
- Dessa forma é evitado que a árvore utilize em excesso uma de suas subárvores;
- Manter a eficiência de operações sobre a estrutura da árvore;

Árvores AVL

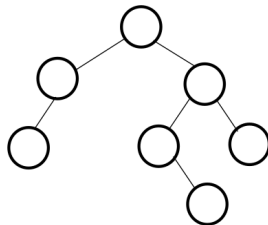
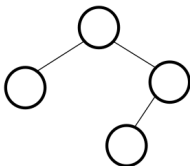
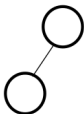
- Criada por “A”derson-“V”elskii e “L”andis em 1962 e publicada no artigo “*An algorithm for the organization of information*”.
- Também é conhecida como árvore balanceada pela altura.
- O balanceamento é feito ao executar uma operação de **inserção** ou **remoção**.

Árvores AVL

- Uma árvore AVL busca manter-se como uma árvore binária **completa** (ou **quase completa**), ou seja, os nós folha sempre estarão no último ou penúltimo nível.
- Não é necessário balancear toda a árvore ao realizar uma operação, e sim na subárvore que ficar desbalanceada.

Árvores AVL

- Em uma árvore AVL, para todos os nós as alturas das suas duas subárvores diferem em, no máximo, uma unidade.
 - **Exemplo:**



Árvores AVL

- Para verificar se uma árvore está balanceada e respeita a restrição descrita anteriormente, é utilizado um conceito de fator de balanceamento (FB);
- O FB de um nó é calculado por:

$$FB = SAE - SAD \quad (1)$$

em que SAE é a altura da subárvore da esquerda e SAD é a altura da subárvore da direita;

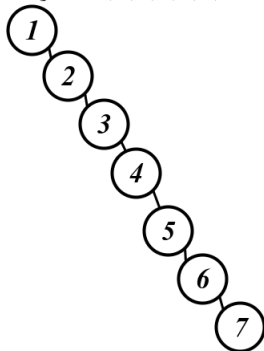
Árvores AVL

- Se o FB de um nó possui o valor -1, 0 ou 1, o nó e suas subárvores estão balanceados;
- Qualquer nó folha possui FB igual à 0;
- Caso contrário, a árvore não é AVL e precisa ser **balanceada**;

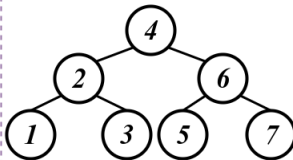
Árvores AVL

- Qual o FB cada nó das árvores:

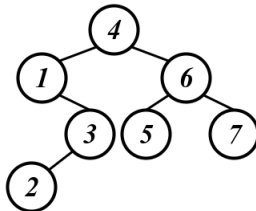
inserção: 1,2,3,4,5,6,7



inserção: 4,2,6,1,3,5,7



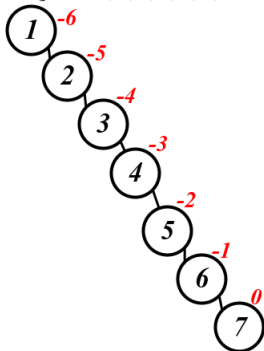
inserção: 4,1,3,6,5,2,7



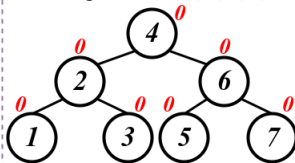
Árvores AVL

- Qual o FB cada nó das árvores:

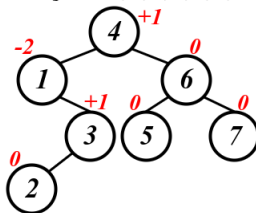
inserção: 1,2,3,4,5,6,7



inserção: 4,2,6,1,3,5,7

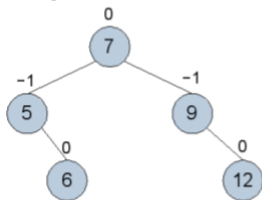


inserção: 4,1,3,6,5,2,7

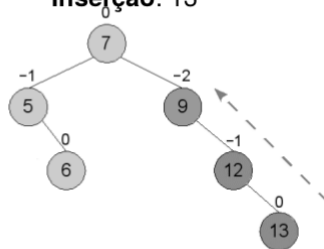


Árvores AVL - Inserção

Inserção: 7, 5, 9, 6, 12

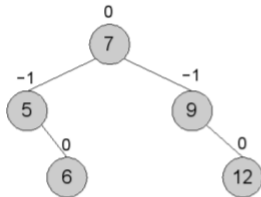


Inserção: 13

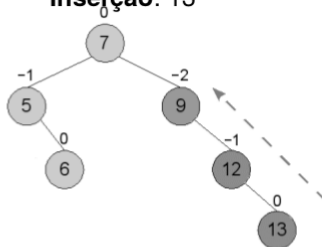


Árvores AVL - Inserção

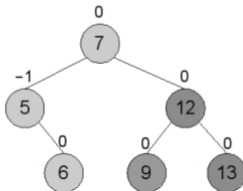
Inserção: 7, 5, 9, 6, 12



Inserção: 13

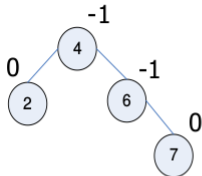


Operação de Balanceamento

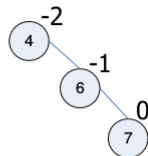


Árvores AVL - Remoção

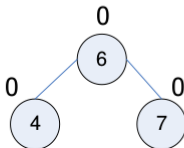
Inserção: 4, 6, 2 e 7.



Remoção: 2



Operação de Balanceamento

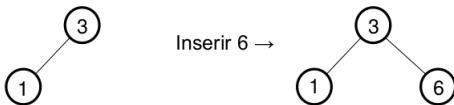


Árvores AVL

- Para manter uma árvore AVL, é necessário aplicar uma transformação na árvore que atenda às seguintes condições:
 - Percurso em-ordem na árvore transformada deve ser igual ao da árvore original.
 - A árvore transformada deve ser **balanceada**.
- O balanceamento é feito através de operações de **rotação**.

Árvores AVL - Inserção

- Inserir um novo nó na árvore.
 - Essa inserção pode ou não violar a propriedade de balanceamento.
 - Caso não viole, operação finalizada.
 - Caso contrário, é necessário restaurar o balanço da árvore através de rotações.



- OBS: É necessário ter um algoritmo de ajuste a cada inserção feita para manter a árvore balanceada.

Árvores AVL - Balanceamento

- Podemos dividir os problemas balanceamento em dois casos:
 - **Caso 1:** o nó raiz de uma sub árvore tem $FB = +2$ (-2) e tem um filho com $FB = +1$ (-1). Ou seja, tanto o FB do pai quanto do filho tem mesmo sinal. Solução: **rotação simples**.
 - **Caso 2:** o nó raiz de uma sub árvore tem $FB = +2$ (-2) e tem um filho com $FB = -1$ ($+1$). Ou seja, os sinais dos FB do pai e do filho são opostos. Solução: **rotação dupla**.

Árvores AVL - Balanceamento

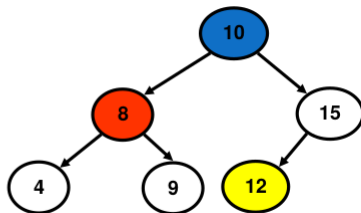
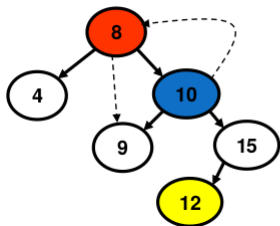
- Quando FB do pai e filho possuem mesmo sinal, a solução utilizada é chamada de **rotação simples** sobre o nó que possui $FB = +2$ (ou -2).
 - Quando o FB é positivo, realizamos uma rotação à direita.
 - Quando o FB é negativo, realizamos uma rotação à esquerda.

Árvores AVL - Balanceamento

- Exemplo 1:

$$\text{FB}(8) = 1 - 3 = -2$$

$$\text{FB}(10) = -1$$

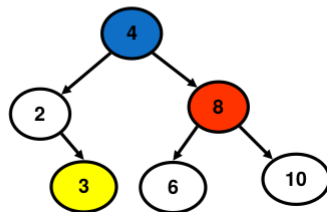
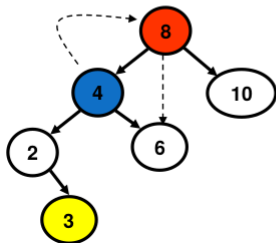


Árvores AVL - Balanceamento

- Exemplo 2:

$$\text{FB}(8) = 3 - 1 = +2$$

$$\text{FB}(4) = +1$$



Árvores AVL - Balanceamento

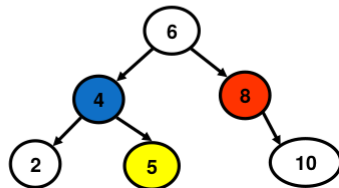
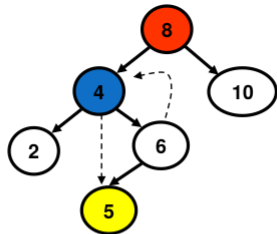
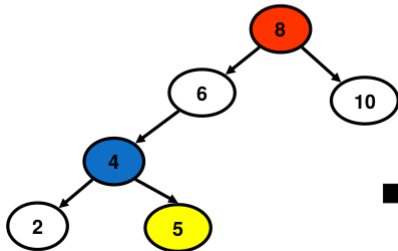
- Quando FB do pai e filho possuem sinais opostos, a solução utilizada é chamada de **rotação dupla**.
 - Primeiro é realizada uma rotação sobre o nó filho ($FB = +1$ ou -1) na direção apropriada:
 - Se FB negativo – rotação à esquerda.
 - Se FB positivo – rotação à direita.
 - Em seguida, a rotação acontece sobre o nó com $FB = +2$ (-2) na direção oposta.

Árvores AVL - Balanceamento

- Exemplo:

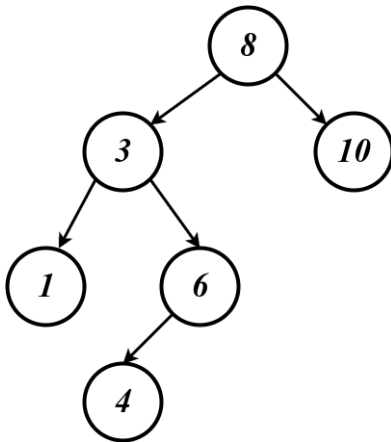
$$FB(8) = 3 - 1 = +2$$

$$FB(4) = -1$$



Exercícios

- Balanceie a árvore:



Bibliografia Básica

- DROZDEK, Adam. Estrutura de Dados e Algoritmos em C++. Editora Pioneira Thomson Learning, 2005.
- Estrutura de dados descomplicada em linguagem C, CAPÍTULO 11 - André Ricardo Backes, <https://www.evolution.com.br/epubreader/estrutura-de-dados-descomplicada-em-linguagem-c-1ed>