Estrutura de Dados

Árvores

Prof. Jorge Luiz Chiara

ÁRVORES

Uma árvore é uma coleção finita de $n \geq 0$ elementos (também chamados de nó, nodo ou vértice).

Se $n = 0$, dizemos que a árvore é nula ou vazia; caso contrário uma árvore apresenta as seguintes características:

- existe um nodo especial denominado **raiz**;
- os demais são particionados em T_1, T_2, ..., T_k estruturas disjuntas denominadas de de subárvores;

A exigência de que as estruturas T_1, T_2, ..., T_k sejam coleções disjuntas, garante que um mesmo nodo não aparecerá em mais de uma subárvore ao mesmo tempo; ou seja, nunca teremos subárvores interligadas. Observe que cada uma das estruturas T_i é organizada na forma de árvore, isto caracteriza uma definição recursiva.

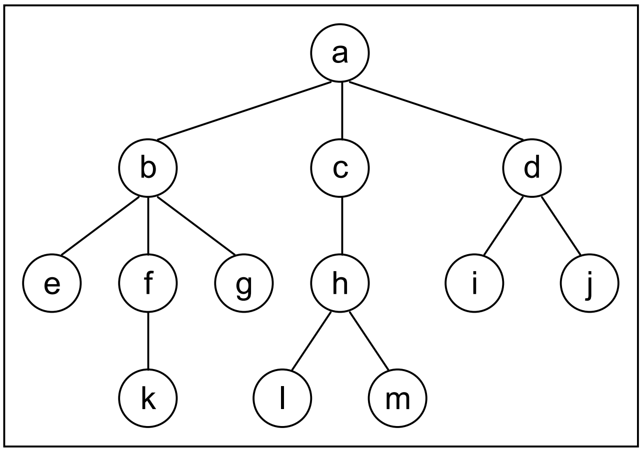


Figura 1: Representação gráfica de uma árvore

O número de subárvores de um nodo denomina-se **grau**. Por exemplo, na Figura 1, o nodo ***a*** tem grau 3, o nodo ***d*** tem grau 2 e o nodo ***c*** tem grau 1. Um nodo que possui grau 0, ou seja, que não possui subárvores, denomina-se terminal ou **folha**, ou nó externo.

São folhas os nodos ***e***, ***k***, ***g***, ***l***, ***m***, ***i*** e ***j***. O grau de uma árvore (n-aridade), é definido como sendo igual ao máximo dos graus de todos os seus nodos. Na ilustração acima, podemos constatar que o grau da árvore é 3, pois nenhum nodo tem mais de 3 subárvores.

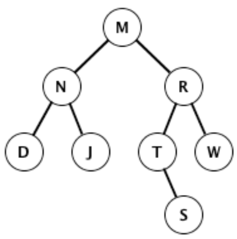
As raízes das subárvores de um nodo denominam-se **filhos** do nodo, que é o **pai** delas. Os filhos do nodo ***b*** são ***e***, ***f*** e ***g***; ***h*** é o pai de ***l*** e ***m***;.

Os filhos do mesmo pai denominam-se **irmãos**. Intuitivamente, pode-se ainda definir termos tais como avô, neto, primo, descendentes, etc.

Por definição, dizemos que a raiz de uma árvore encontra-se no nível 1 (alguns autores adotam como sendo zero). Estando um nodo no nível n, seus filhos estarão no nível n + 1. A **altura** de uma árvore é definida como sendo o máximo dos níveis de todos os seus nodos. Uma árvore nula tem altura 0. A árvore da figura acima tem altura igual a 4. A subárvore com raiz ***d*** tem altura 2.

Exercício: Dada a árvore ao lado, indique:

- Os nós folha
- Grau da árvore
- Altura da árvore
- Os filhos de R



ÁRVORES BINÁRIAS

Uma **árvore binária** é uma árvore de grau máximo igual a dois (2), cujas subárvores estão ordenadas, ou seja, uma árvore binária, pode ser nula, ou então tem as seguintes características:

- existe um nodo especial denominado **raiz**;
- os demais nodos são particionados em T_1 e T_2 estruturas disjuntas de árvores binárias, onde T_1 é denominada subárvore esquerda e T_2, subárvore direita da raiz.

Árvore binária é um caso especial de árvore em que nenhum nodo tem grau superior a 2, isto é, nenhum nodo tem mais que dois filhos. Adicionalmente, para árvores binárias existe um "ordem de posição", ou seja, distingue-se entre uma subárvore esquerda e uma direita. Por exemplo, as árvores binárias representadas na figura abaixo são consideradas distintas: a primeira tem a subárvore direita nula e a segunda, a esquerda.

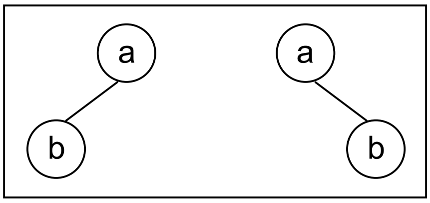


Figura 4.2 – Duas árvores binárias distintas

A terminologia básica utilizada para árvores binárias é a mesma definida para árvores de grau genérico. Podemos dizer agora filho esquerdo, filho direito, etc.

Exercício: Quantas árvores binárias com 3 nodos podemos desenhar?

PERCURSO EM ÁRVORES BINÁRIAS

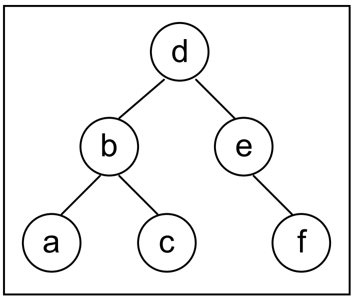
Percorrer ou atravessar uma árvore binária significa “processar de forma sistemática, cada um de seus nodos. Existem quatro tipos básicos de atravessamento:

- em nível (não trataremos em nosso conteúdo)
- pré-ordem - R-E-D
- em-ordem - E-R-D
- pós-ordem – E-D-R

A sequência de acesso R-E-D, do percurso pré-ordem, significa “processar a raiz, “chamar recursivamente” o percurso para a subárvore esquerda e, em seguida, “chamar recursivamente” o percurso para a subárvore direita.

De forma análoga, construímos os percursos em-ordem e pós-ordem.

Exemplo, vamos mostrar os percursos em-ordem, pré-ordem e pós-ordem para a figura abaixo.



Uma árvore binária com subárvores

a) pré-ordem (R-E-D): d, b, a, c, e, f
b) em-ordem (E-R-D): a, b, c, d, e, f
c) pós-ordem (E-D-R): a, c, b, f, e, d

Exercício: Uma arvore binária possui os seguintes percursos:
em ordem: #, $, +, /, >, &, @, X, Y, %, V
pré-ordem: /, $, #, +, X, &, >, @, %, Y, V.
Desenhe a árvore correspondente e o percurso pós-ordem.

REPRESENTAÇÃO DE EXPRESSÕES ARITMÉTICAS ATRAVÉS ARVORES BINÁRIAS

Uma operação é dita binária se possui 2 operandos e, unária caso tenha somente um operando. Uma notação é a forma que representamos, por exemplo, uma operação aritmética. A notação infixa é aquela em que a representação mostra o operador entre os operandos; prefixa, onde o operador está antes dos operandos e pós-fixa, quando o operador está colocado após os operandos.

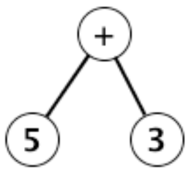
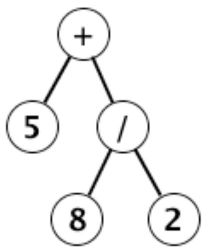
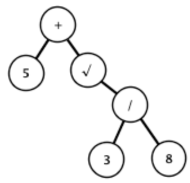
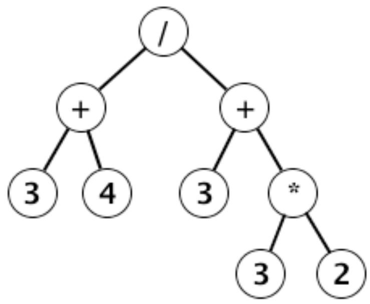
Exemplos:
a) 5+3: infixa
b) +53: prefixa
c) 53+: pós-fixa

Para representar uma expressão aritmética em uma árvore binária devemos adotar a seguinte regra:

- todo operador é uma raiz na árvore;
- todo operando é uma folha;
- as operações seguem as regras de precedência na ordem natural da esquerda para direita.

A notação infixa corresponde ao percurso em-ordem.
A notação prefixa corresponde ao percurso pré-ordem.
A notação pós-fixa corresponde ao percurso pós-ordem.

Exemplos:

a) Expressão: 5+3	b) Expressão: 5 + 8 / 2	b) Expressão: $5+\sqrt{3/8}$	c) Expressão: (3+4) / (3+3*2)
 infixa: 5+3 prefixa: +53 pós-fixa: 53+	 infixa: 5+8/2 prefixa: +582 pós-fixa: 582/+	 infixa: 5+√ 3/8 prefixa: +5√ /32 pós-fixa: 538/√ +	 infixa: 3+4/3+3*3 prefixa: /+34+3*32 pós-fixa: 34+332*+/

Exercícios: Dadas expressões, abaixo, mostre as árvores binárias que as representam e as notações correspondentes (infixa, prefixa e pós-fixa) em cada caso.

a) A / (C + D) + (J * Q)
b) 3 * (5 + 2) + 2*(5-4)
c) A+B/C-D+E+F
d) 8/2+8*4+32*(5+2)
e) $\dfrac{-b+\sqrt{b*b-4ac}}{2a}$

ÁRVORES DE BUSCA BINÁRIA

Definição: É uma árvore binária, cuja raiz armazena o elemento R, é denominada **árvore de busca binária** se:

- todo elemento armazenado na subárvore esquerda é menor que R;
- nenhum elemento armazenado na subárvore direita é menor que R;
- as subárvores esquerda e direita também são árvores de busca binária.

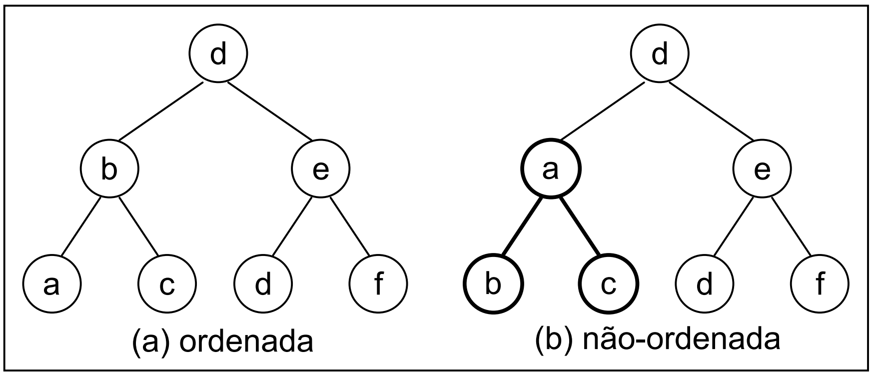


Figura 4.3 – Árvores binárias

Observe que a definição de árvore de busca binária também é recursiva. A árvore representada na figura (b) acima, tem o elemento ***d*** na raiz, todos os elementos na subárvore esquerda (***a***, ***b*** e ***c***) são menores que a raiz; nenhum elemento da subárvore direita (***d***, ***e*** e ***f***) é menor que a raiz e, no entanto, ela não é uma árvore de busca binária. Isto é devido ao fato de a subárvore esquerda não ser uma árvore de busca binária. Note que a subárvore esquerda, cuja raiz é ***a***, tem uma subárvore esquerda não ser uma árvore de busca binária. Note que a subárvore esquerda, cuja raiz é ***a***, tem uma subárvore esquerda que armazena um elemento maior que a raiz. Logo, a árvore como um todo não está de acordo com a definição e não é considerada, portanto, uma árvore de busca binária!

Exercício: Dada uma árvore de Busca Binária (ABB) inicialmente vazia, insira e desenhe os seguintes elementos e nessa ordem: 13, 6, 19, 4, 10, 16, 21, 1, 5, 8. Em seguida, escreva os percursos: Pré-ordem, Em-ordem e Pós-ordem.

ÁRVORE AVL

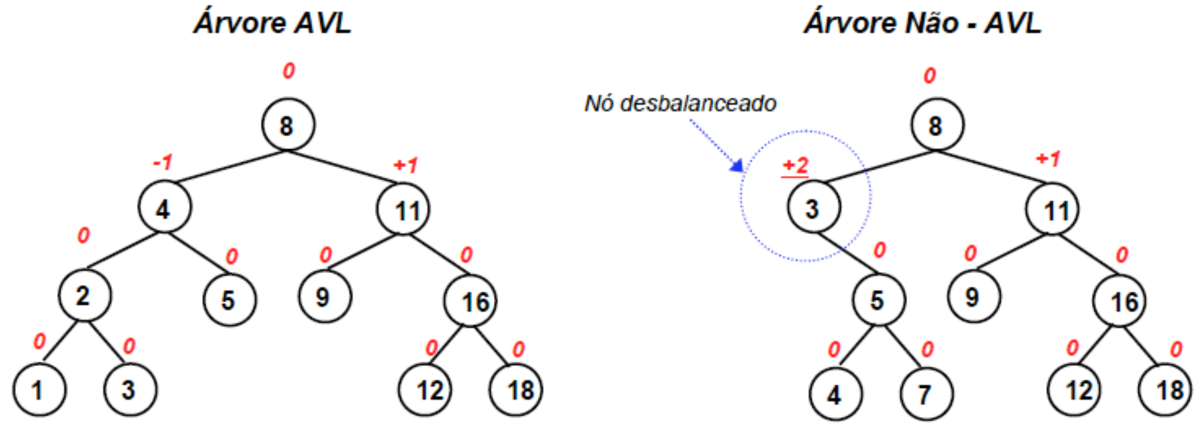
G.M. Adelson-Velskii e E.M. Landis (1962) apresentaram uma árvore de busca binária que é balanceada com respeito a altura das subárvores.

Definição: Uma árvore AVL é:

1) Uma árvore binária de Busca (BST)
2) Uma árvore balanceada cuja diferença em altura entre a subárvore esquerda e a subárvore direita (em qualquer nó) deve ser, no máximo, em valor absoluto, menor ou igual a 1, ou seja $|FB \leq 1|$, onde FB significa a diferença entre as alturas entre as subárvores esquerda e direita.

Numa árvore AVL, cada chave é única e as operações de inserção ou remoção tentam manter o equilíbrio da árvore.

Exemplos:

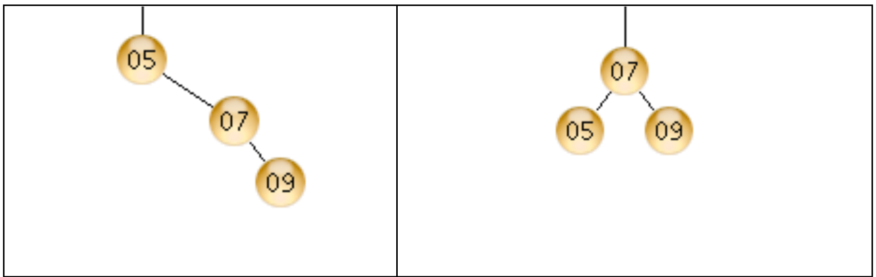


Rotações em Árvores Binárias

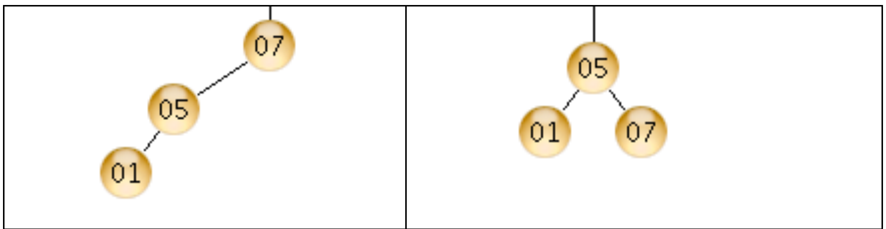
Rotacionar dois nós em árvores binária permite a permutação da nó raiz em uma subárvore qualquer. Podemos realizar dois tipos de rotações:

1) Rotação Simples
   a. à esquerda
   b. à direita
2) Rotação dupla
   a. Simples à esquerda seguida de uma simples à direira
   b. Simples à direita seguida de uma simples à equerda.

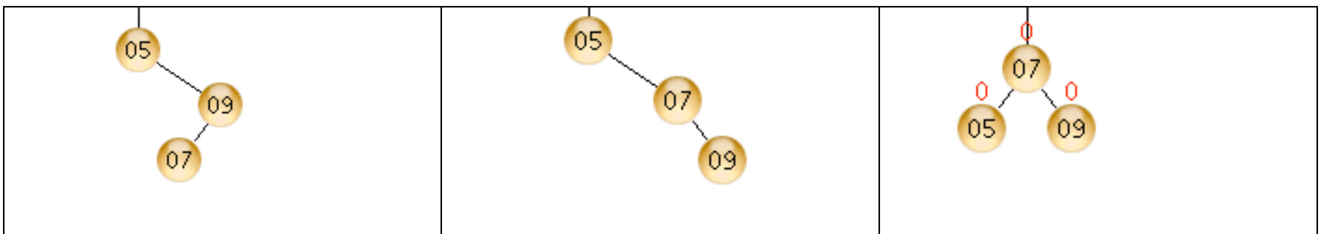
Abaixo, seguem alguns exemplos de rotações em árvores binárias.



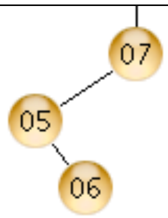
Exemplo de simples rotação à esquerda:

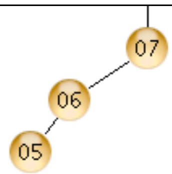


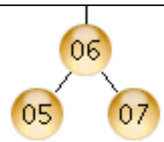
Exemplo de simples rotação à direita:



Exemplo de dupla rotação à direita seguida de uma à esquerda:







Exemplo de dupla rotação à esquera seguida de uma à direita:

Exercício: Considerando uma Árvore AVL, inicialmente vazia, desenhe os seguintes elementos, nessa ordem: 2,4,6,8, 9, 11, 13 e 17

Exercício: Considerando uma Árvore AVL, inicialmente vazia, desenhe os seguintes elementos, nessa ordem: 15, 13, 11, 9, 7, 6, 4,3, 2e 1

Exercício: Considerando uma Árvore AVL, inicialmente vazia, desenhe os seguintes elementos, nessa ordem: 7, 6, 5, 9, 11, 8, 3, 15, 19, 13 e 10.

Observação: para estudos, utilize o link:
http://webdiis.unizar.es/asignaturas/EDA/AVLTree/avltree.html