



Estruturas Avançadas de Dados I

Árvores

1

Introdução

- As listas encadeadas usualmente fornecem maior flexibilidade do que as matrizes, mas são estruturas **lineares** e é difícil usá-las para organizar uma **representação hierárquica**;
- Listas lineares como **pilhas** (LIFO) e **filas** (FIFO) possuem propósitos específicos, restringindo ainda mais o seu uso;
- É necessário uma estrutura de dados que possa ter **múltiplas relações** de forma a estabelecer uma **hierarquia**, além de realizar uma busca eficiente.



2

O que são árvores?

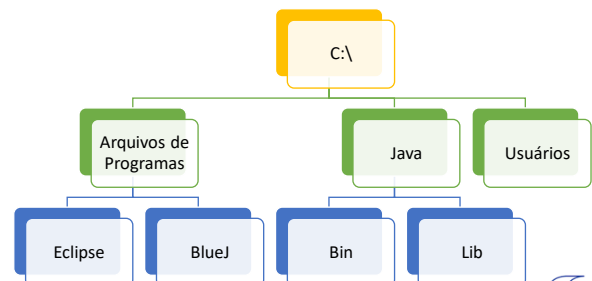
- Acima de tudo, são **estruturas de dados** adequadas para a representação de informações de forma **hierárquica**;
- Uma árvore enraizada T , ou simplesmente uma árvore, é um conjunto finito de elementos denominados nós ou nodos tais que:
 - $T = 0$ é a árvore dita vazia ou
 - existe um nó especial r , chamado raiz de T ; os restantes constituem um único conjunto vazio ou são divididos em m (deve ser maior ou igual a 1) conjuntos distintos não vazios que são as subárvores de r , cada subárvore a qual é, por sua vez, uma árvore.
 - Cada nó v de T que não é a raiz tem apenas um pai.



3

O que são árvores?

- Além disso, similar às listas lineares e array, árvore é um **modelo abstrato** de uma estrutura hierárquica;
- Consiste de nós com a relação pai-filho.



4

Porque utilizar árvores?

- Diversas aplicações necessitam de **estruturas mais complexas** que as listas estudadas até agora;
- **Inúmeros problemas** podem ser modelados através de árvores;
- Árvores admitem um **tratamento computacional eficiente** quando comparadas às estruturas mais genéricas como os grafos (os quais, por sua vez são mais flexíveis e complexos).

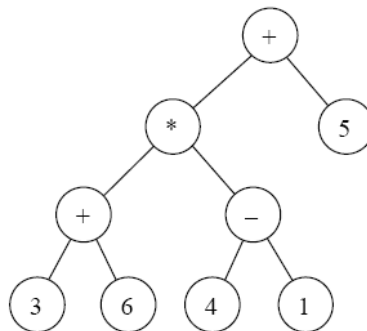


5

Tipos de Estruturas em Árvore

- Representação de uma expressão aritmética

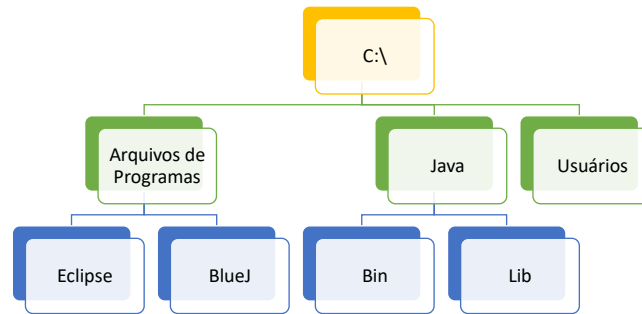
$(3+6)*(4-1)+5$



6

Tipos de Estruturas em Árvore

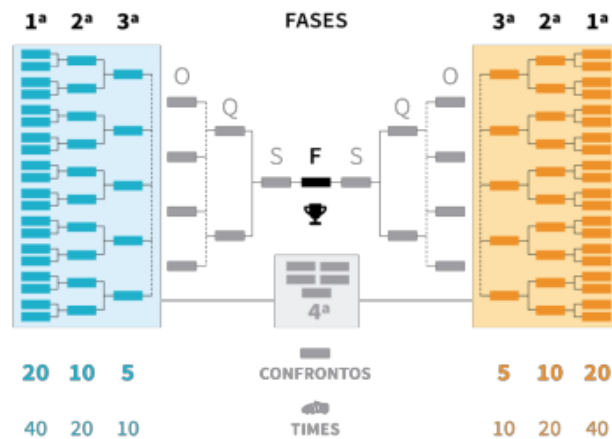
- Diretório de um Sistema Operacional



7

Tipos de Estruturas em Árvore

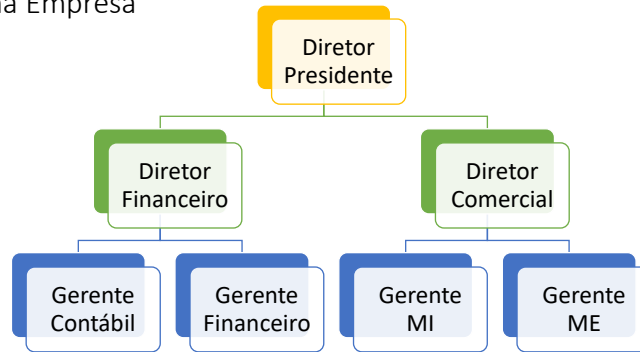
- “Mata-mata” da Libertadores



8

Tipos de Estruturas em Árvore

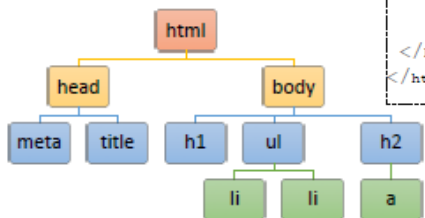
- Organograma de uma Empresa



9

Tipos de Estruturas em Árvore

- Uma página HTML



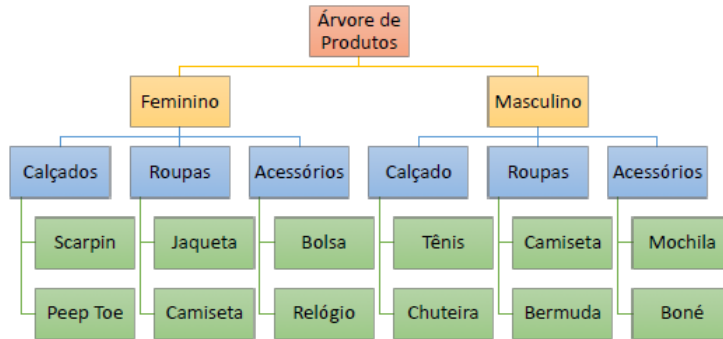
```

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Simples</title>
  </head>
  <body>
    <h1>Web site</h1>
    <ul> <li>Lista de itens 1</li> <li>Lista de itens 2</li> </ul>
    <h2><a href="http://www.unisinos.br">Unisinos</a></h2>
  </body>
</html>
  
```

10

Tipos de Estruturas em Árvore

- Estrutura de Produto



Conceitos Básicos

- **Nó ou nodo:** elemento da árvore;
- **Arco:** ligação entre os nós;
- **Raiz:** primeiro nó da árvore;
- **Sub-árvores:** são os galhos da árvore;
- **Grau de saída de um nó:** número de sub-árvores de um nó;
- **Grau de uma árvore:** é o máximo entre os graus de seus nós;
- **Folha (ou nó terminal):** um nó que possui grau zero, ou seja, não possui sub-árvores;
- **Filhos:** são as raízes das sub-árvores de um nó;
- **Nó não terminal:** é um nó que não é uma folha e é diferente da raiz;

Conceitos Básicos

- **Pai:** é o primeiro nó no caminho do nó n para a raiz;
- **Irmão:** são nós que possuem o mesmo pai;
- **Ancestral:** qualquer nó no caminho da raiz até o nó em questão;
- **Descendente:** qualquer nó a partir dos caminhos possíveis do nó em questão;
- **Caminho:** é a sequência única de arcos que conectam os nós;
- **Comprimento:** número de arcos de um caminho;
- **Altura (ou Profundidade):** é o nível do nó folha que tem o mais longo caminho até a raiz somando 1;
- **Nível:** é a sua distância da raiz da árvore. A raiz tem nível 0.



13

Conceitos Básicos

- O número de filhos permitido por nó e as informações armazenadas em cada nó diferenciam os **diversos tipos de árvores** existentes;
- Todos os nós **são acessíveis** a partir da raiz;
- Existe um **único caminho** entre a raiz e qualquer outro nó.
- Cada nó possui um pai, exceto a raiz.



14

Conceitos Básicos

- A raiz é um nó que não possui antecessores;
- As folhas não possuem nós filhos, ou seus filhos são estruturas vazias;
- Cada nó tem que ser atingível a partir da raiz através de uma sequência única de arcos.



15

Conceitos Básicos

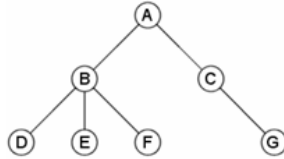
- **Árvore Genérica:** cada nó poderá ter n subárvores, ou seja, não tem um número máximo de subárvores;
- **Árvore N-ária:** cada nó tem no máximo n subárvores. Há um caso especial que estudaremos na próxima aula onde $n = 2$.



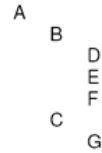
16

Representação

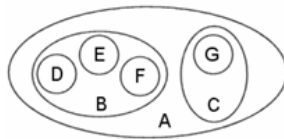
- Grafo (representação mais utilizada)



- Identação



- Diagrama de Venn (ou digrama de inclusão)



- Parênteses Aninhados

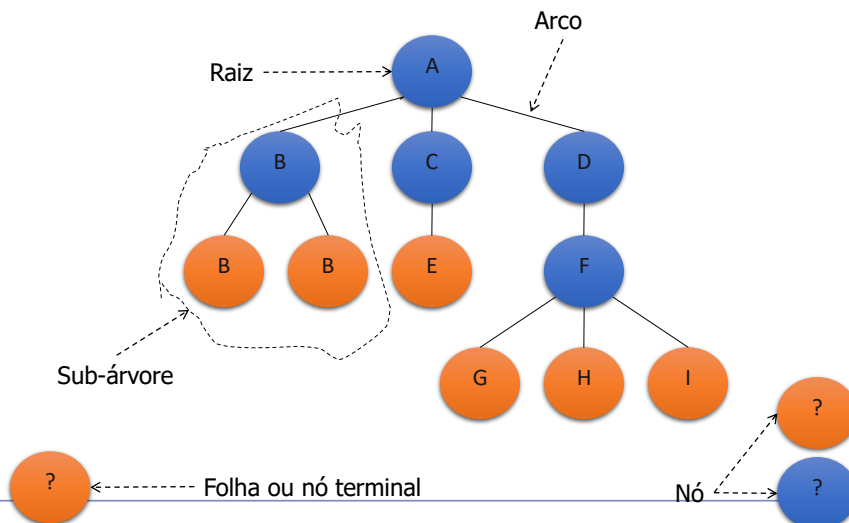
(A (B(D, E, F), C(G)))

- Representações físicas através de arrays e encadeamentos.



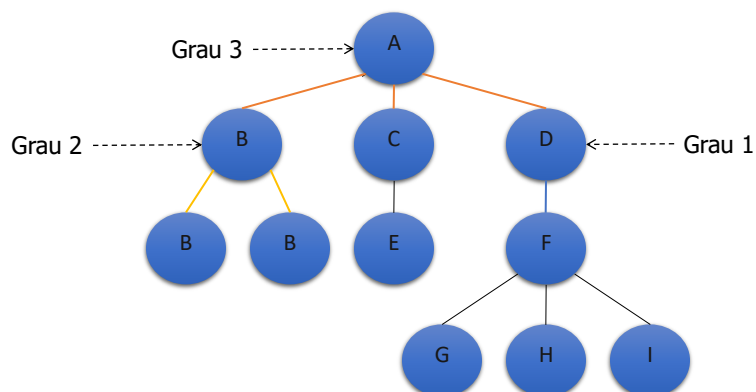
17

Aplicação dos Conceitos



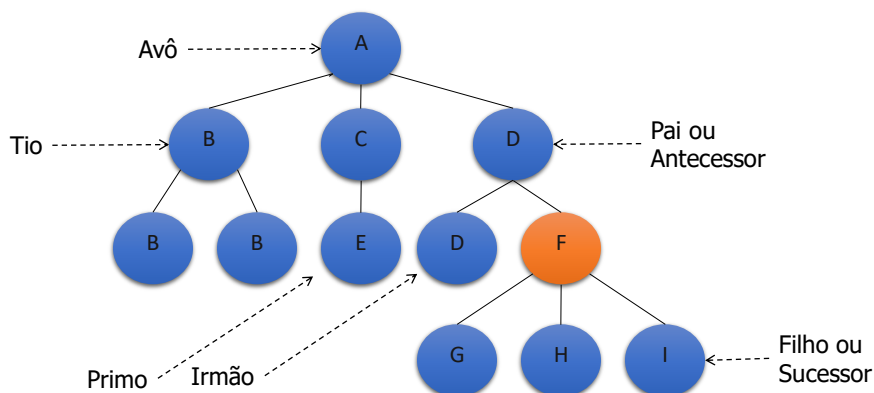
18

Aplicação dos Conceitos - Grau



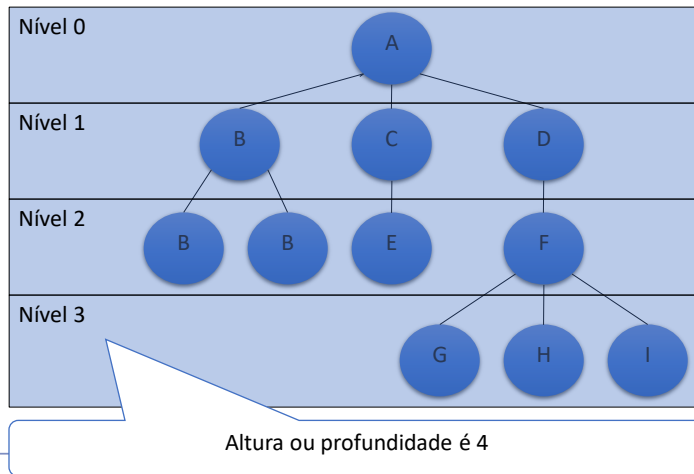
19

Aplicação dos Conceitos - Parentesco



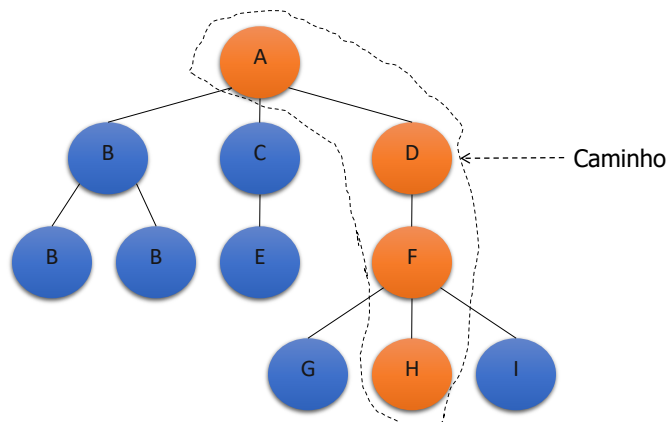
20

Aplicação dos Conceitos – Nível/Altura



21

Aplicação dos Conceitos - Caminho



22

O que iremos trabalhar em árvores?

- Criação da árvore (estrutura de dados)
- Inserção de um novo nó
- Exclusão de um nó
- Acesso ao nó
 - Tipos de caminhamento (percurso)
- Exclusão da árvore



23

Percurso em Árvores

- O percurso em árvores é o processo de visitar cada nó da árvore exatamente uma vez;
- O percurso pode ser interpretado como colocar todos os nós em uma linha;
- Mas qual a ordem? Existem $n!$ percursos diferentes, quase todos caóticos;
- Os básicos são percurso em **profundidade** e percurso em **amplitude**.

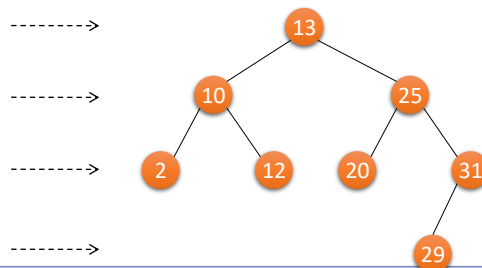


24

Percurso em Árvores - Extensão

- Um percurso em amplitude (extensão ou nível) é visitar cada nó começando do menor nível e move-se para os níveis mais altos, nível após nível, visitando cada nó da esquerda para a direita:

- Breadth - First Search (BFS)



Percurso: 13, 10, 25, 2, 12, 20, 31, 29

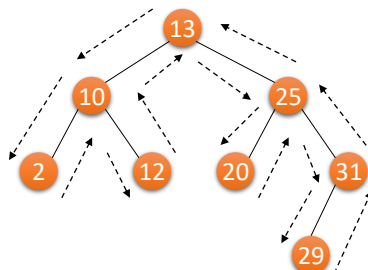


25

Percurso em Árvores - Profundidade

- O percurso em profundidade prossegue tanto quanto possível à esquerda (ou direita), então se move para trás até a primeira encruzilhada, vai um passo para a direita (ou esquerda) e novamente, tanto quanto possível, para a esquerda (ou direita).

- Depth - First Search (DFS)



V – Visitar um nó
L – Percorrer à esquerda
R – Percorrer à direita

VLR VRL
LVR RVL
LRV RLV

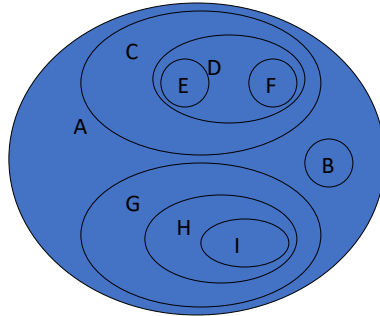
Percurso: 2, 12, 10, 20, 29, 31, 25, 13



26

Exercícios

- 2.1 - Para as representações de árvore abaixo, apresente a árvore em grafo.



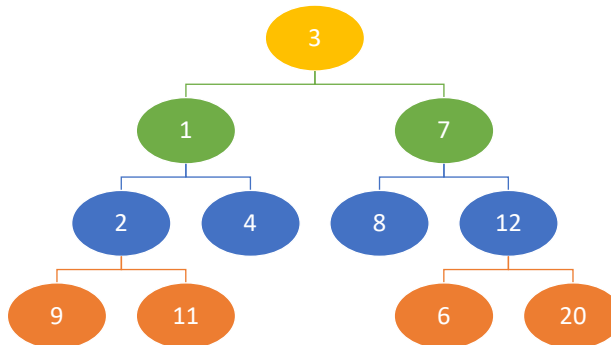
(A (B (X, Y (M, N (O))), Z), C (D, E (F, G, H))))



27

Exercícios

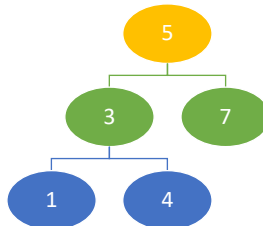
- 2.2 – Apresente todos os conceitos vistos em aula para a seguinte árvore:



28

Exercícios

- 2.3 – Uma das características da utilização de uma árvore é a inserção ordenada (veremos mais adiante em árvores binárias). No lado esquerdo do nó, temos o filho com o menor elemento e, no lado direito do nó, temos o filho com o maior elemento, conforme ilustração abaixo.



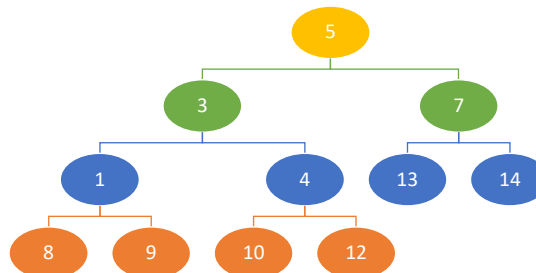
- A árvore em questão teve a inserção dos seus elementos na seguinte ordem: 5, 3, 1, 4, 7.
- Crie uma árvore com a inserção dos seguintes elementos, assumindo que essa seja ordenada: A, B, C, D, E, F, G, 2, 3.



29

Exercícios

- 2.4 – Quando falamos em percurso (caminhamento), temos diversos algoritmos para profundidade e extensão.



- Apresente o percurso em profundidade;
- Apresente o percurso em extensão.



30

Referências Bibliográficas

- ASCENCIO, A. F. G; ARAÚJO, G. S. **Estruturas de Dados**. São Paulo: Pearson Prentice Hall, 2010. 432 p.
- BATISTA, C. A. T. **Estruturas de Dados**. Lâminas segundo semestre, 2009.
- CORMEN, Thomas H. et al. **Introduction to algorithms**. 3. ed. Cambridge: MIT, 2009. xix. 1292 p.



31

Prof. Márcio Garcia Martins
marciog@unisinios.br

Para anotar: *ao enviar e-mail sempre coloque o seguinte prefixo no assunto*
[EADI-ano-semester] – Nome do aluno



32