

Estruturas Avançadas de Dados I (Árvores)

Prof. Gilberto Irajá Müller

Introdução

- As listas encadeadas e array são estruturas **lineares**. Portanto, é difícil usá-las para organizar uma **representação hierárquica** ou até mesmo fazer uma simples busca;
- 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.

O que são árvores?

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

O que são árvores? (cont.)

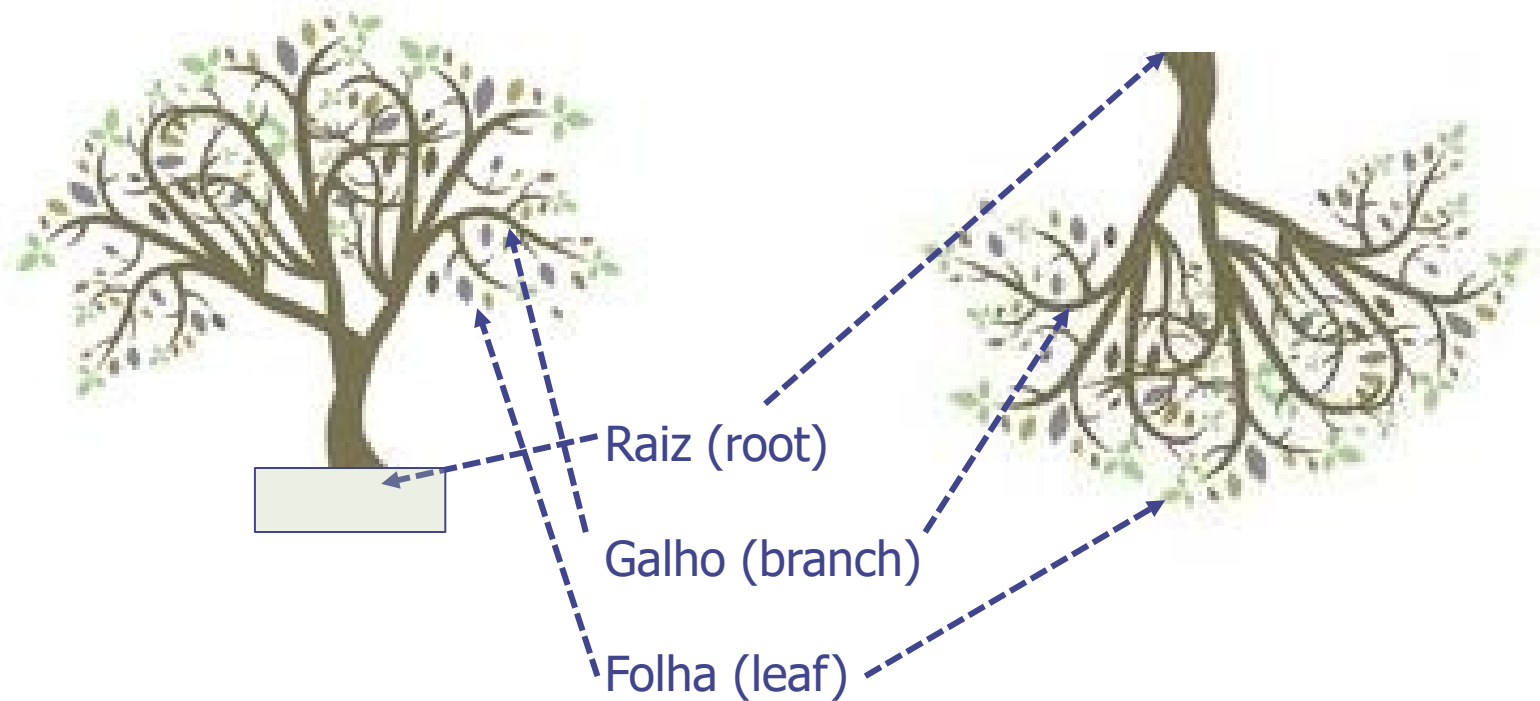
- Além disso, similar às listas lineares e array, árvore é um **modelo abstrato** de uma estrutura hierárquica;
- Consiste de nós com a relação pai-filho.



Visão de Árvore

Natural

Estrutura de Dados



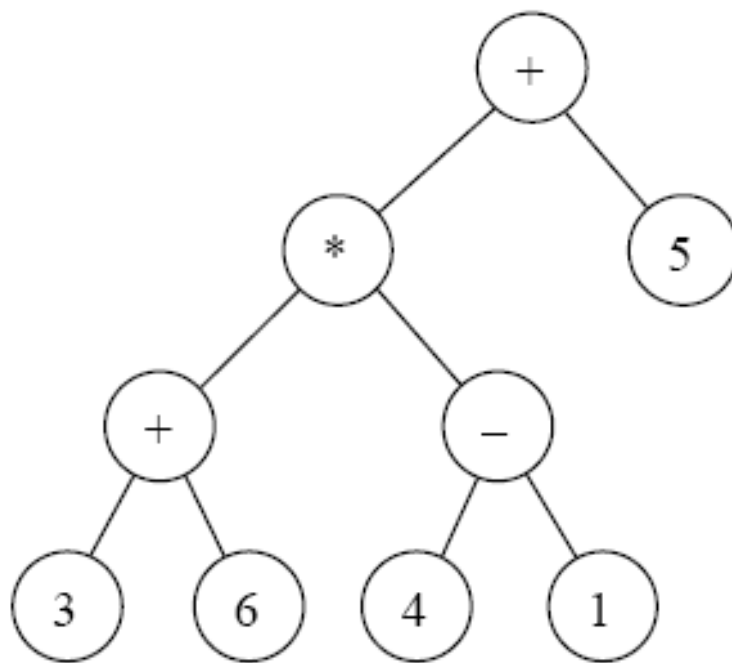
Por que utilizar árvores?

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

Aplicação de Estruturas em Árvore

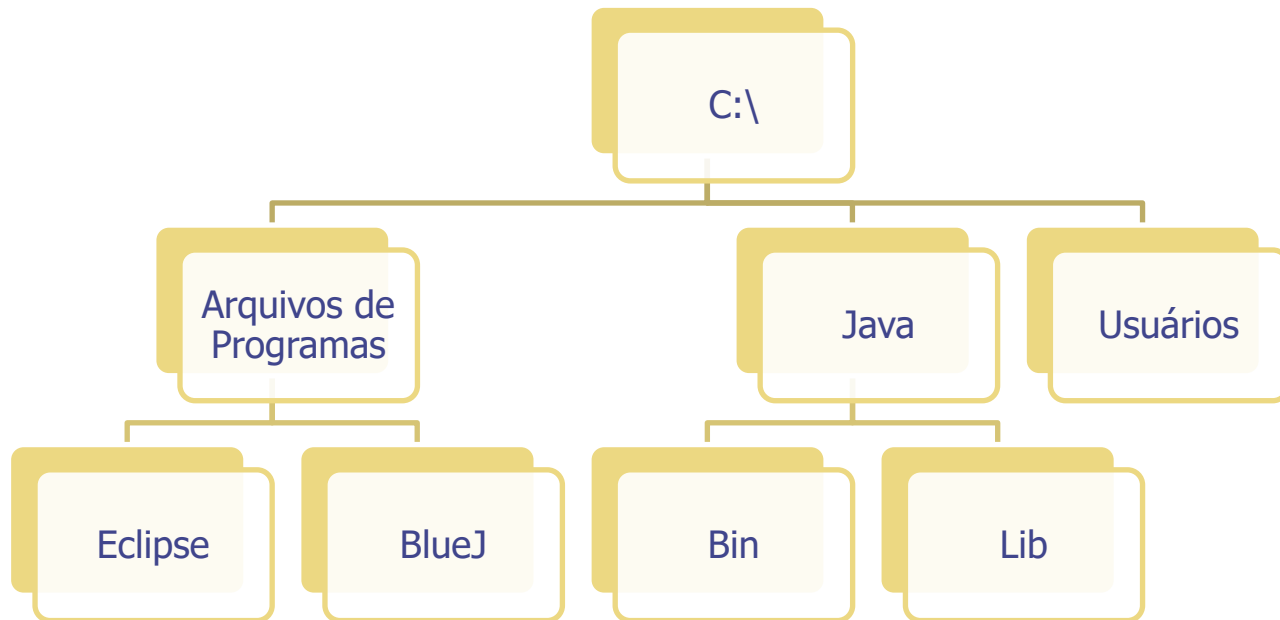
- Representação de uma expressão aritmética

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



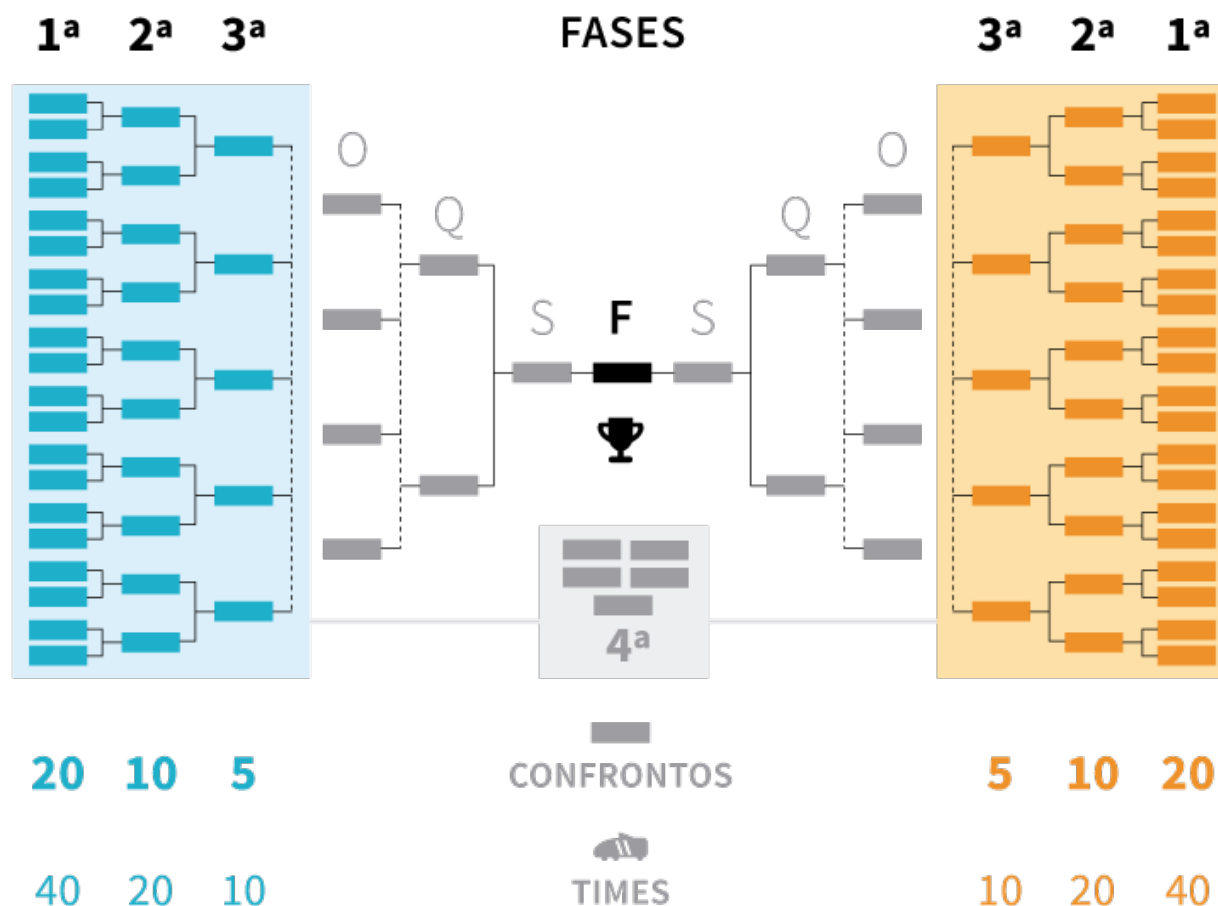
Aplicação de Estruturas em Árvore (cont.)

- Diretório de um Sistema Operacional



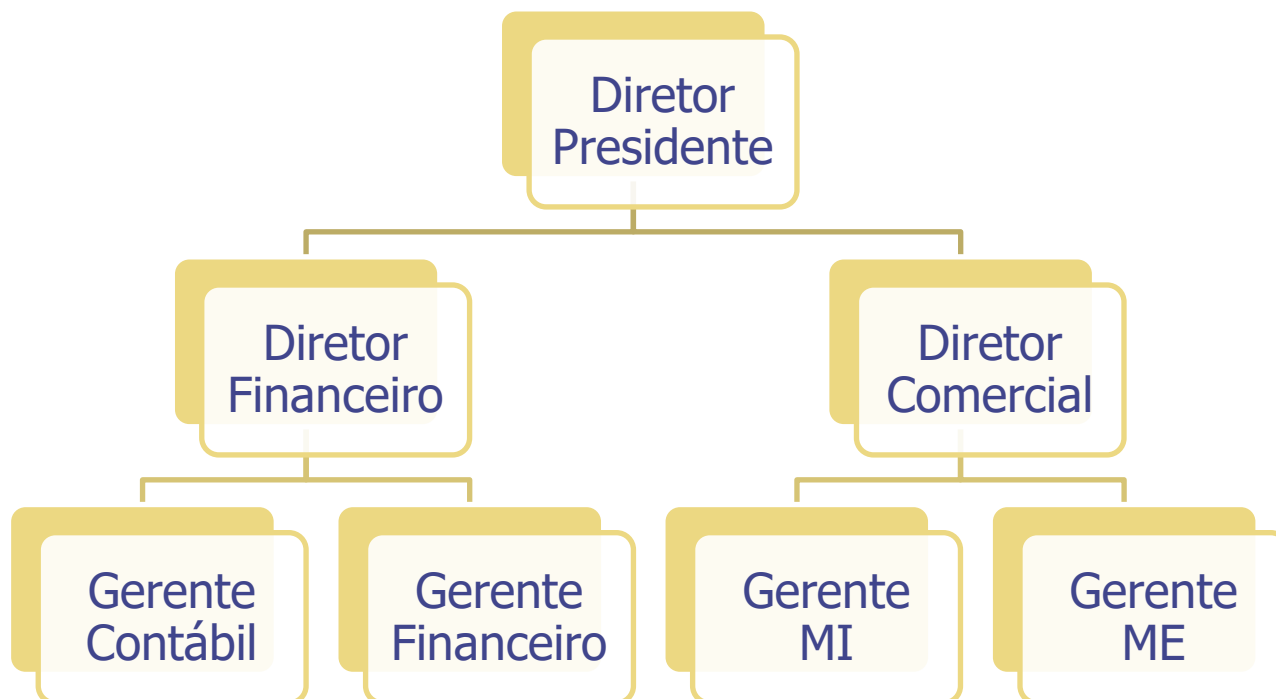
Aplicação de Estruturas em Árvore (cont.)

- “Mata-mata” da Libertadores



Aplicação de Estruturas em Árvore (cont.)

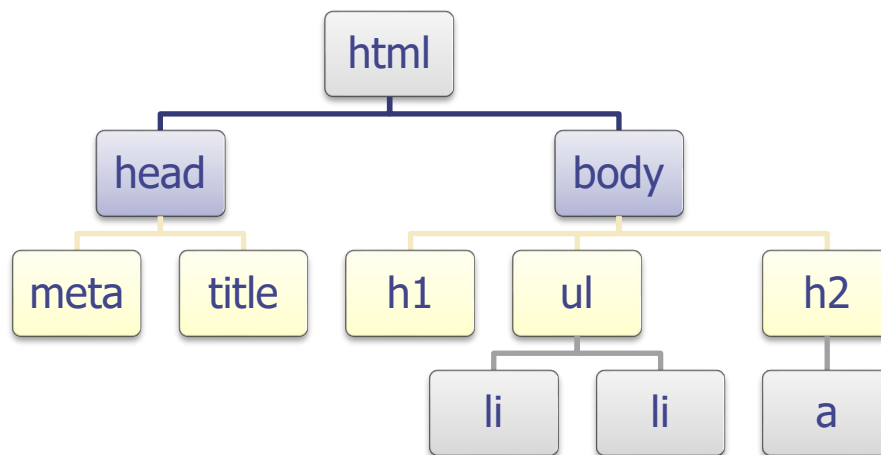
- Organograma de uma Empresa



Aplicação de Estruturas em Árvore (cont.)

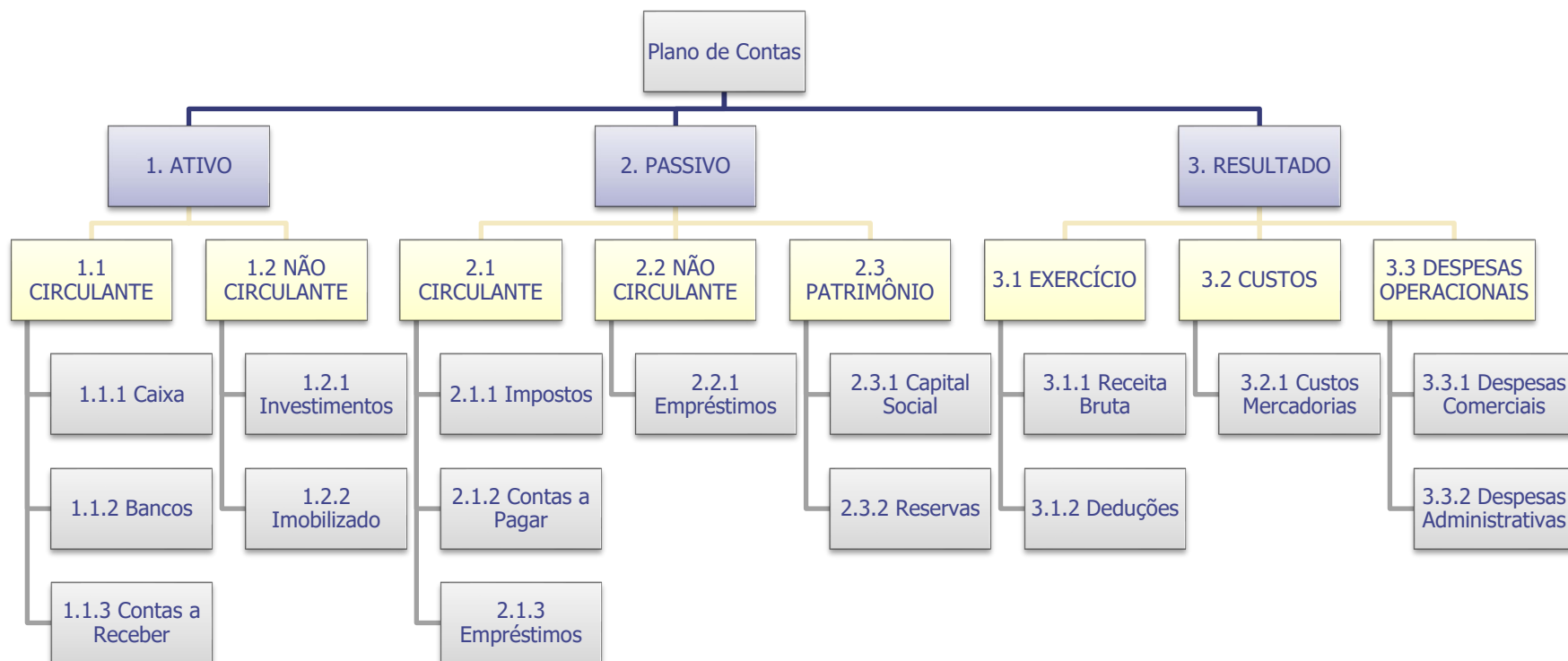
- Uma página HTML

```
<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>
```



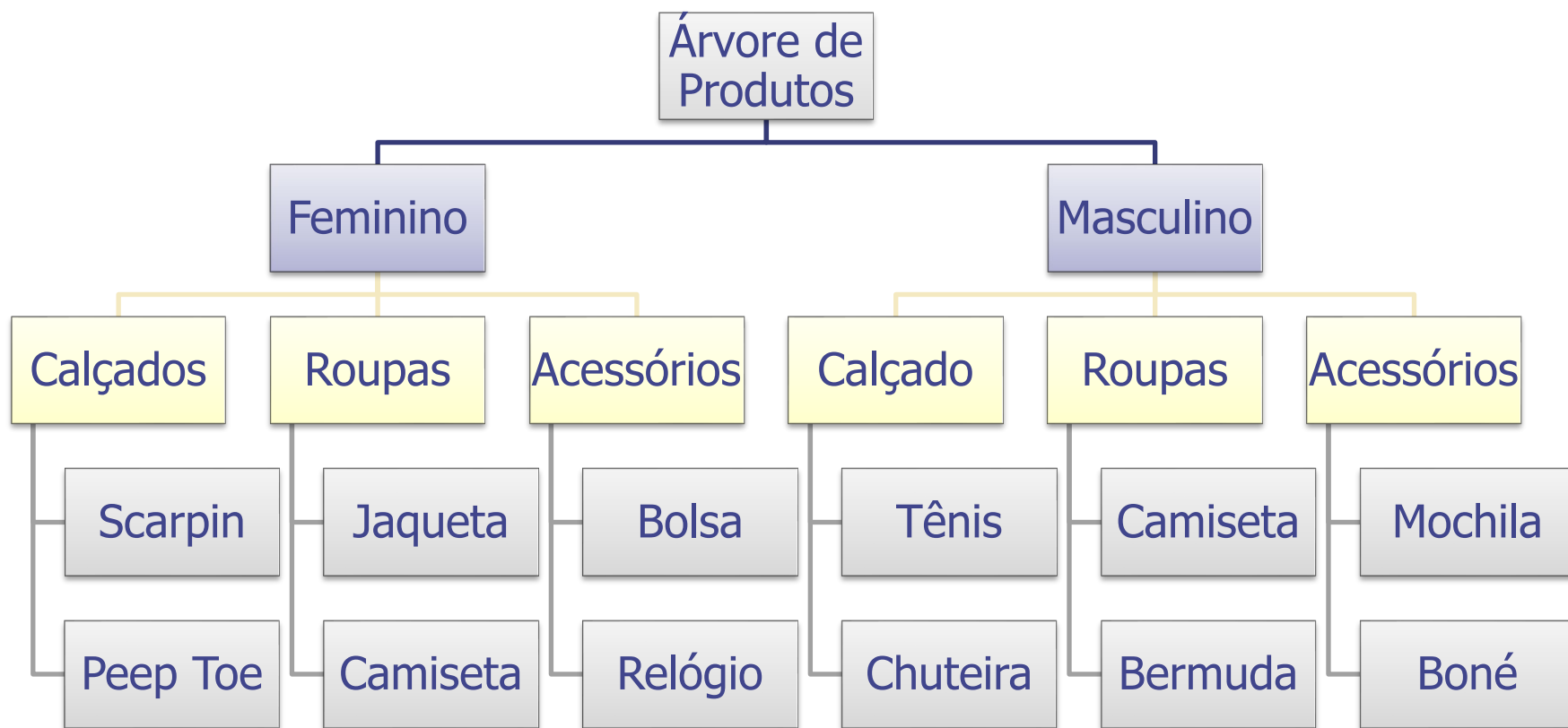
Aplicação de Estruturas em Árvore (cont.)

- Plano de Contas Contábil



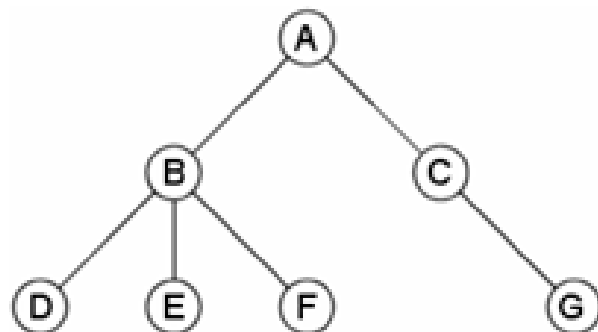
Aplicação de Estruturas em Árvore (cont.)

- Estrutura de Produto

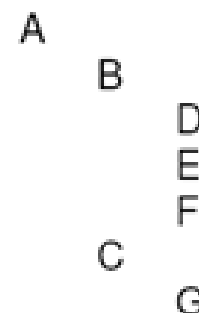


Representação de Árvores

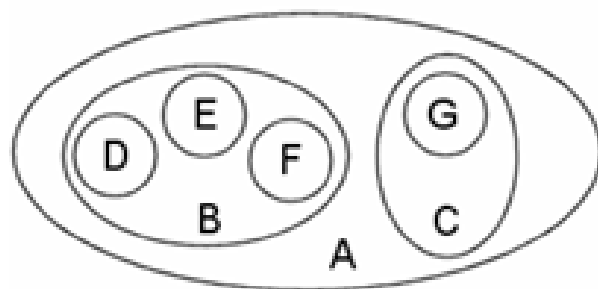
- Grafo (representação mais utilizada)



- Identação



- Diagrama de Venn (ou digrama de inclusão)



- Parênteses Aninhados

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

- Representações físicas através de arrays e encadeamentos.

Conceitos Básicos – Terminologia

- **Nó ou nodo (node)**: elemento da árvore;
- **Arco (edge)**: ligação entre dois nós;
- **Raiz (root)**: primeiro nó da árvore;
- **Subárvore (subtree)**: são os galhos da árvore; consiste de um nó e seus descendentes;
- **Grau de um nó (degree of a node)**: número de subárvores de um nó ou o número de filhos;
- **Grau de uma árvore (degree of a tree)**: é o número máximo entre os graus de seus nós;
- **Folha (ou nó terminal ou nó externo) (external node or leaf)**: um nó que possui grau zero, ou seja, não possui subárvores;
- **Nó não terminal (ou nó interno) (internal node)**: é um nó que não é uma folha e é diferente da raiz (cuidado: alguns autores tratam a raiz como um nó interno, pois depende do propósito! Em nosso contexto, a raiz não faz parte!!!).

Conceitos Básicos – Terminologia (cont.)

- **Pai (parent):** o pai de um nó n é um nó que possui caminho para o nó n , sendo o primeiro nó no caminho do nó n para a raiz;
- **Irmão (sibling):** são nós que possuem o mesmo pai;
- **Filho (child):** é a raiz da subárvore de um nó;
- **Ancestral (ancestor):** qualquer nó no caminho da raiz até o nó em questão (inclusive).
- **Descendente (descendent):** qualquer nó a partir dos caminhos possíveis do nó em questão (inclusive).

Conceitos Básicos – Terminologia (cont.)

- **Caminho (path):** é a sequência única de arcos que conectam os nós;
- **Comprimento (path length):** número de arcos de um caminho;
- **Profundidade (depth):** é o número de ancestrais do nó (excluindo ele) ou o comprimento da raiz até o nó; recursão para baixo ou para cima;
- **Altura (height):** é o maior comprimento do nó até uma folha qualquer; recursão para baixo; é comum relacionar profundidade com o nó e altura com a árvore, porém, não há impedimento em aplicar as terminologias para a árvore e o nó;
- Profundidade e altura são simétricas, ou seja, a altura de uma árvore T é a maior profundidade de uma das folhas;
- **Nível (level):** o nível d é um nó em profundidade d .

Conceitos Básicos – Terminologia (cont.)

- O número de filhos permitidos 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;
- À exceção da raiz, cada nó possui um pai.

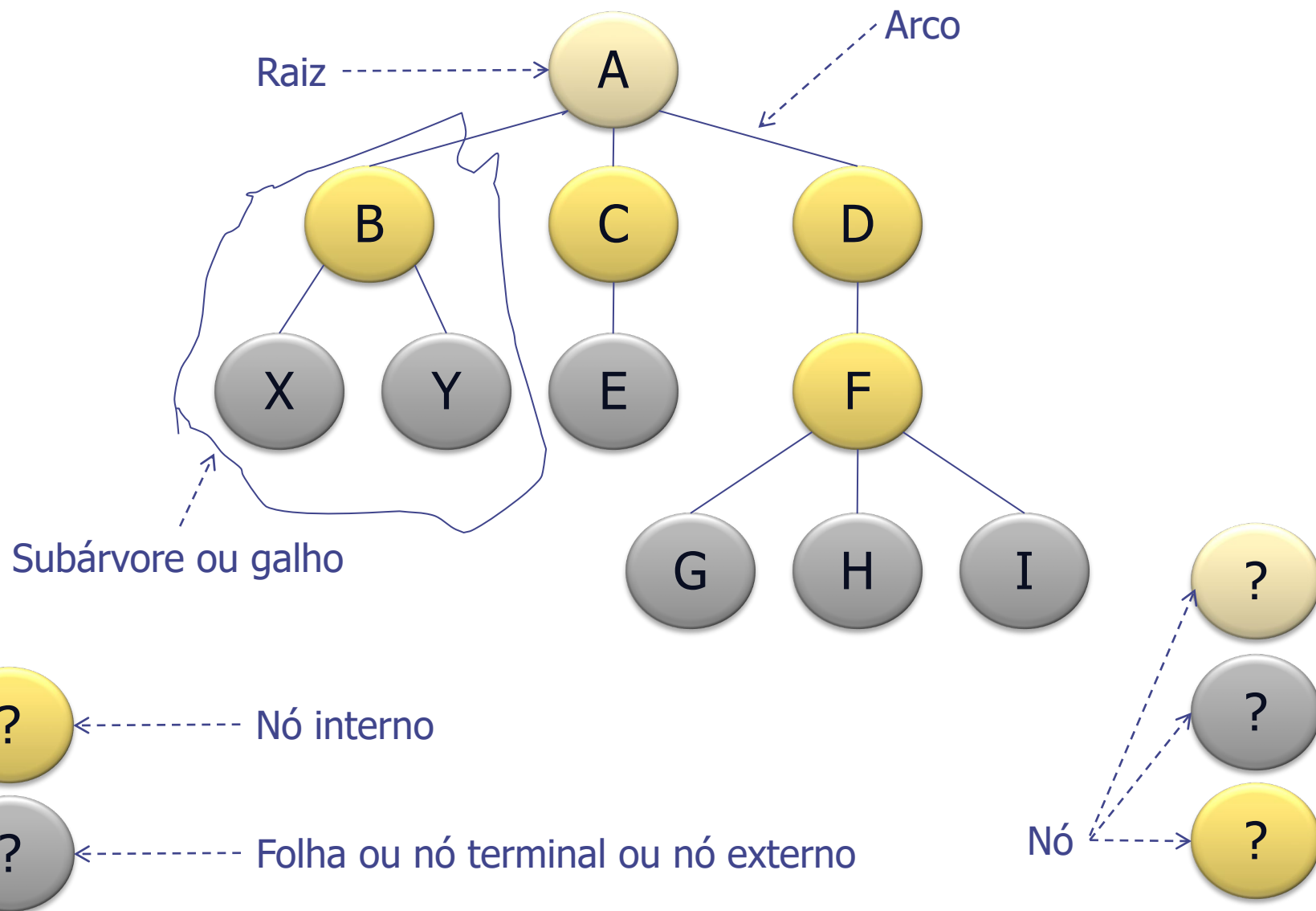
Conceitos Básicos – Terminologia (cont.)

- A raiz é um nó que não possui antecessores (similar ao bullet anterior!);
- 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.

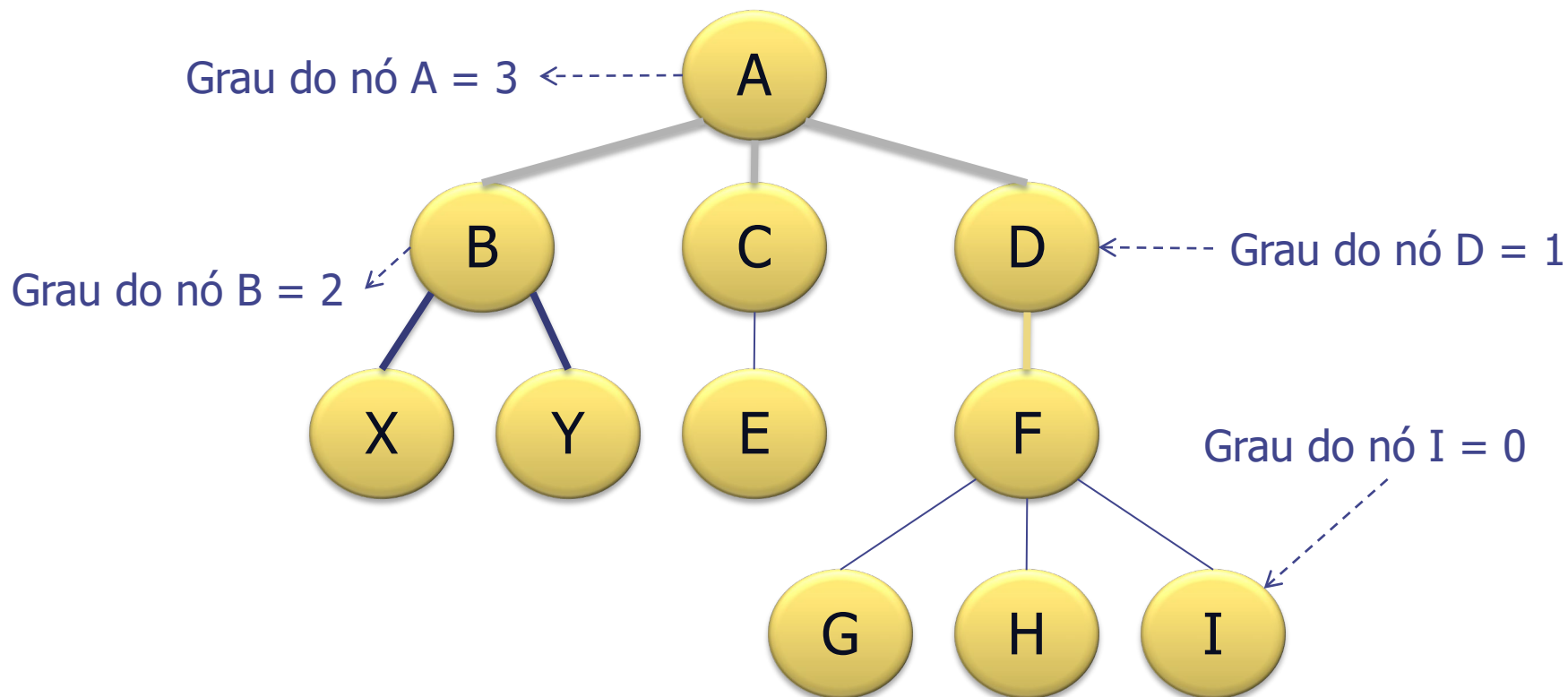
Conceitos Básicos – Terminologia (cont.)

- **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 $n = 2$.

Aplicação dos Conceitos



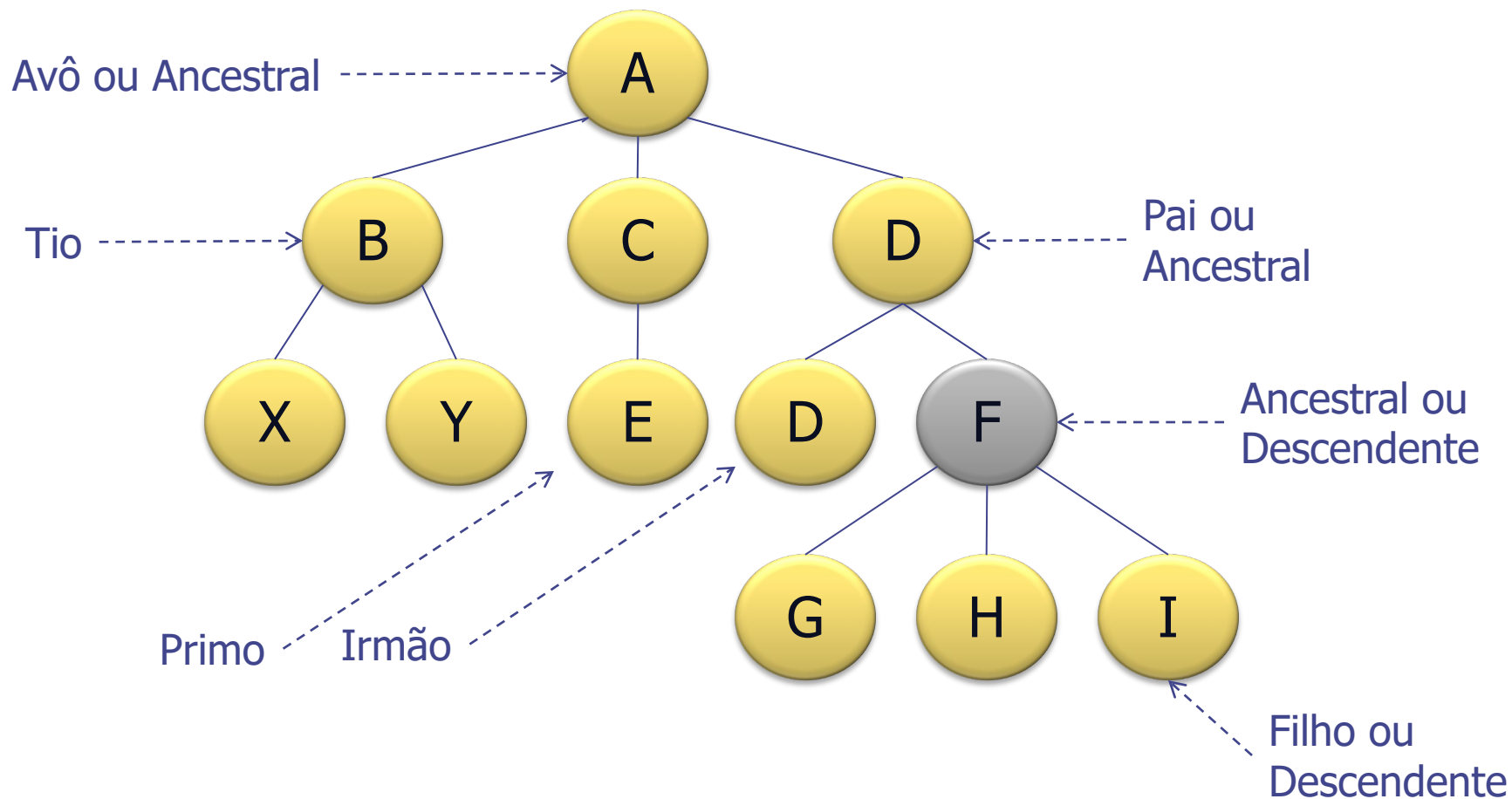
Aplicação dos Conceitos - Grau



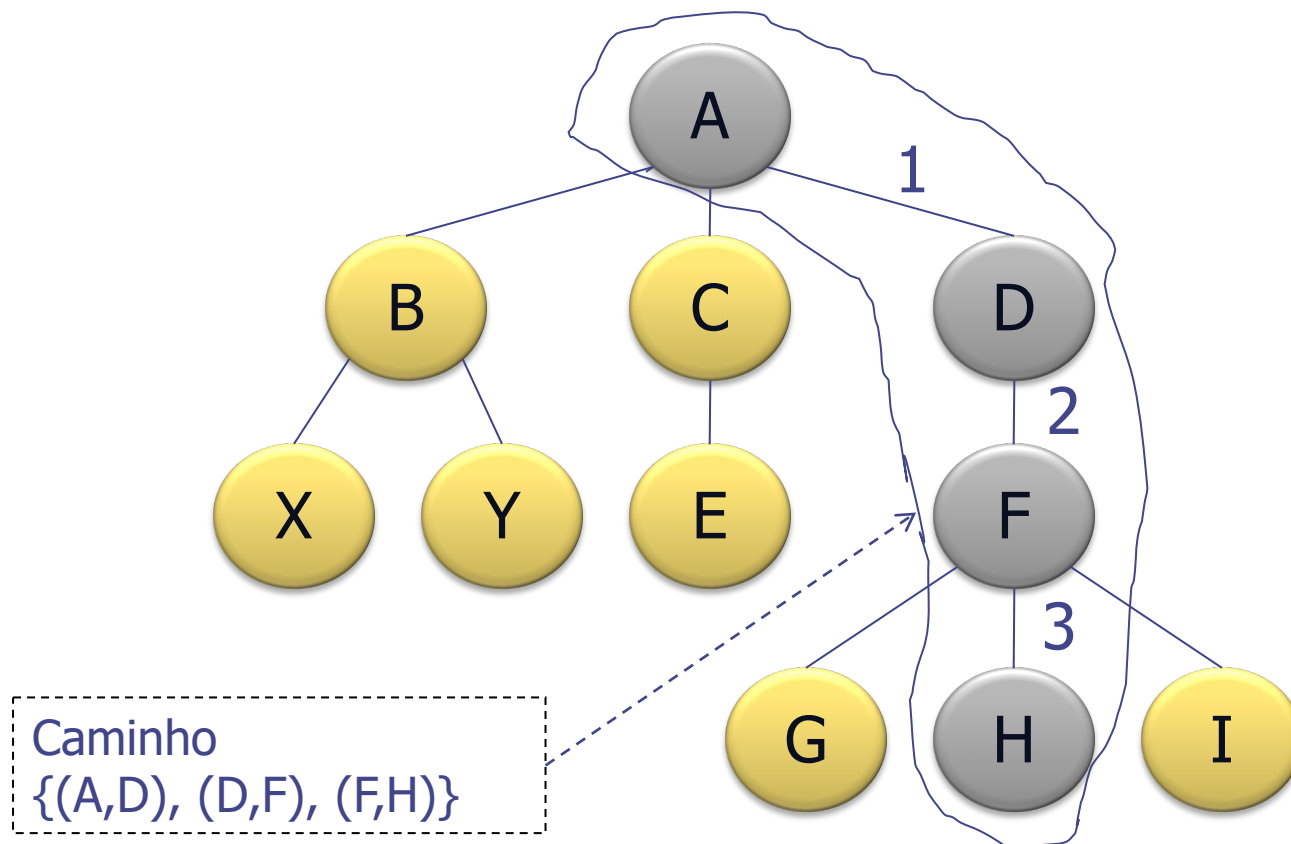
Grau da árvore = 3 (A e F)

Grau de um nó (degree of a node): número de subárvores de um nó ou número de filhos;
Grau de uma árvore (degree of a tree): é o número máximo entre os graus de seus nós.

Aplicação dos Conceitos - Parentesco



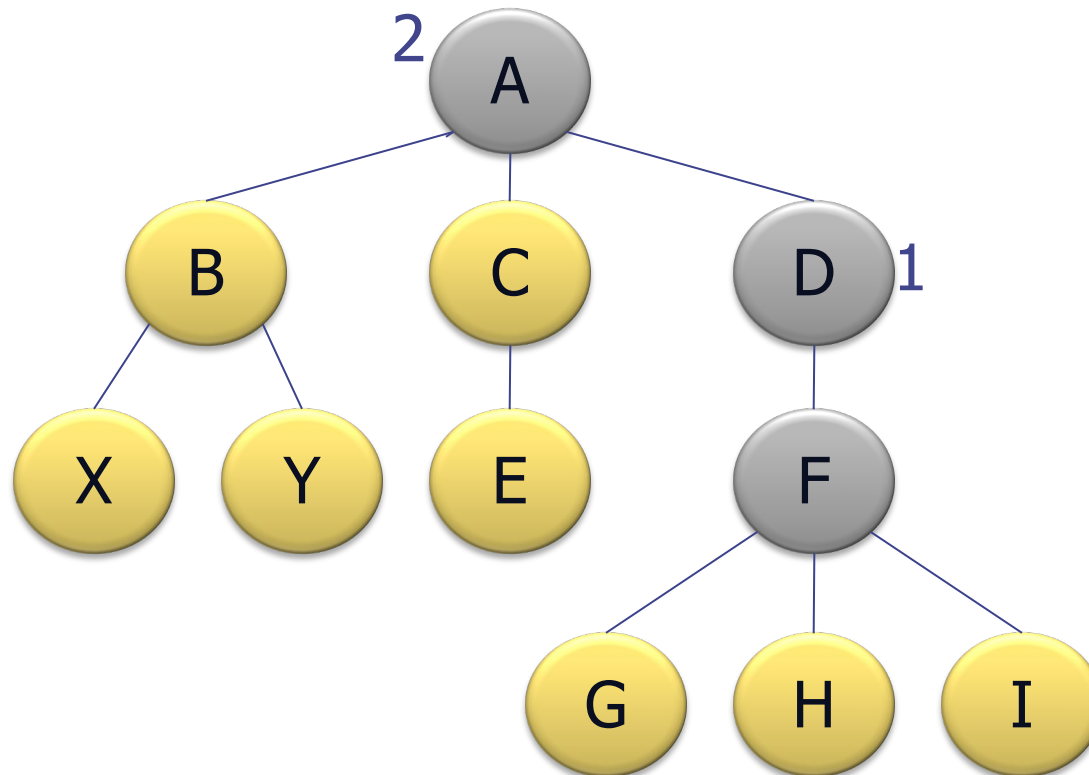
Aplicação dos Conceitos - Caminho



Comprimento do caminho (A,H) é 3

Comprimento (path length): número de arcos de um caminho.

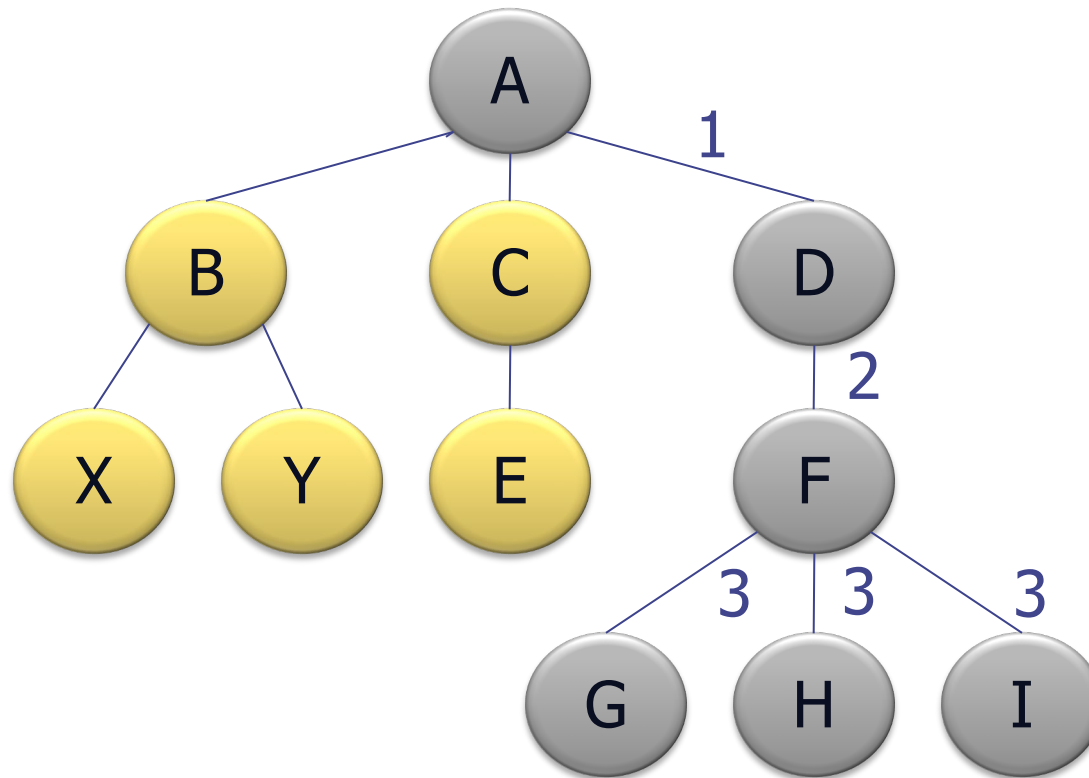
Aplicação dos Conceitos - Profundidade



Profundidade do nó F é 2

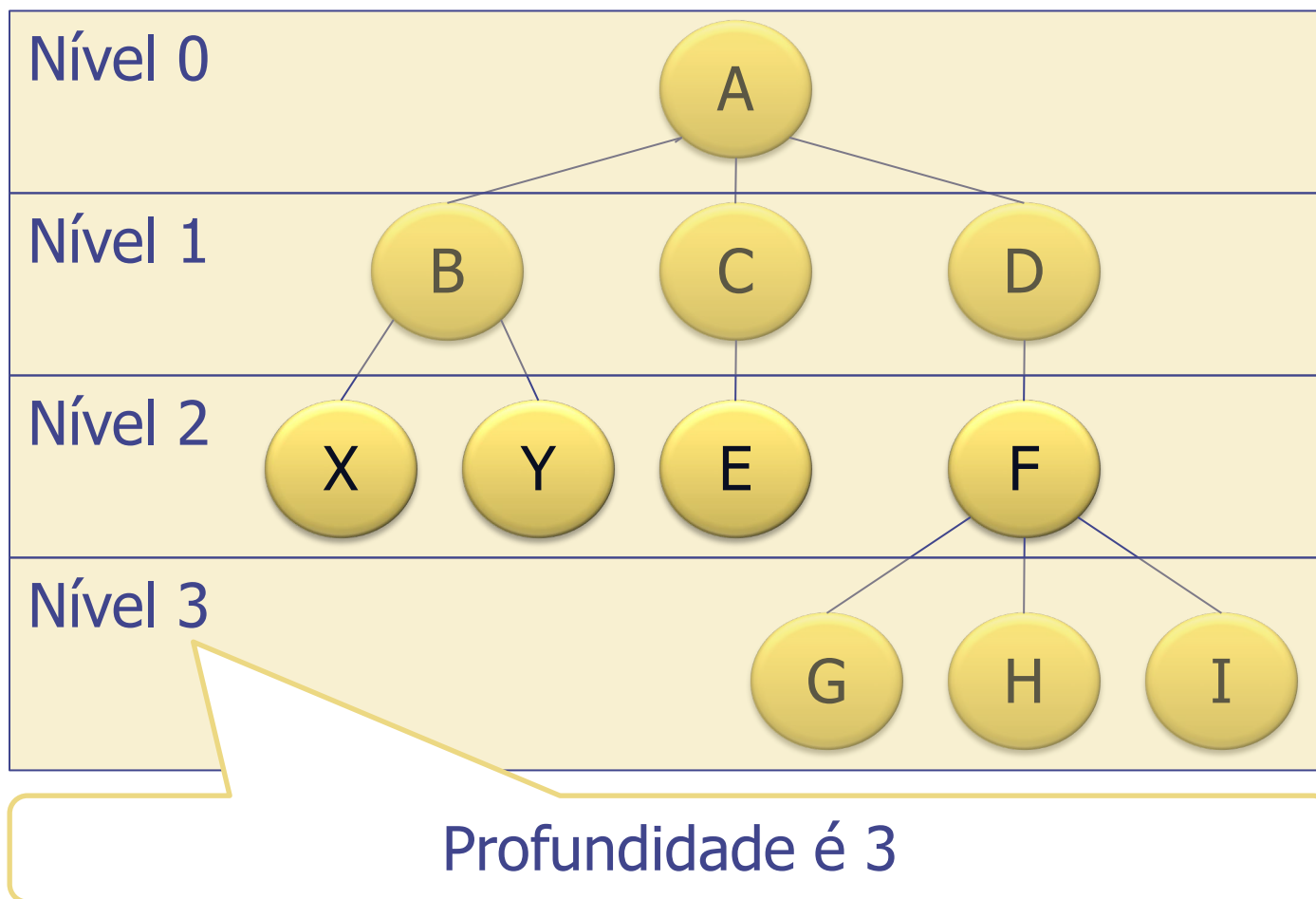
Profundidade (depth): é o número de ancestrais do nó (excluindo ele) ou o comprimento da raiz até o nó.

Aplicação dos Conceitos - Altura



Altura (height): é o maior comprimento do nó até uma folha qualquer.

Aplicação dos Conceitos – Nível



Nível (level): o nível d é um nó em profundidade d .

O que iremos trabalhar em árvores?

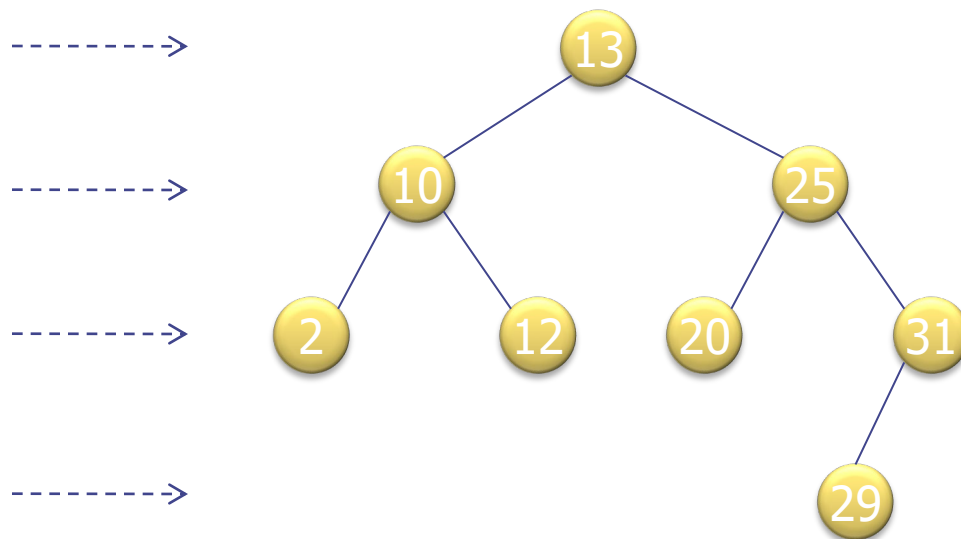
- Criação da árvore (estrutura de dados)
- Inserção de um nó
- Exclusão de um nó
- Acesso ao nó
 - Tipos de caminhamento (percurso)
- Exclusão da árvore

Percurso em Árvores

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

Percurso em Árvores - Amplitude

- Um percurso em amplitude (extensão ou nível) consiste em visitar cada nó começando do menor nível (raiz) movendo-se para os níveis mais altos, nível após nível, visitando cada nó da esquerda para a direita:
 - Breadth First Search (BFS)



Fila: 13

Fila: 10, 25

Fila: 25, 2, 12

Fila: 2, 12, 20, 31

Fila: 12, 20, 31

Fila: 20, 31

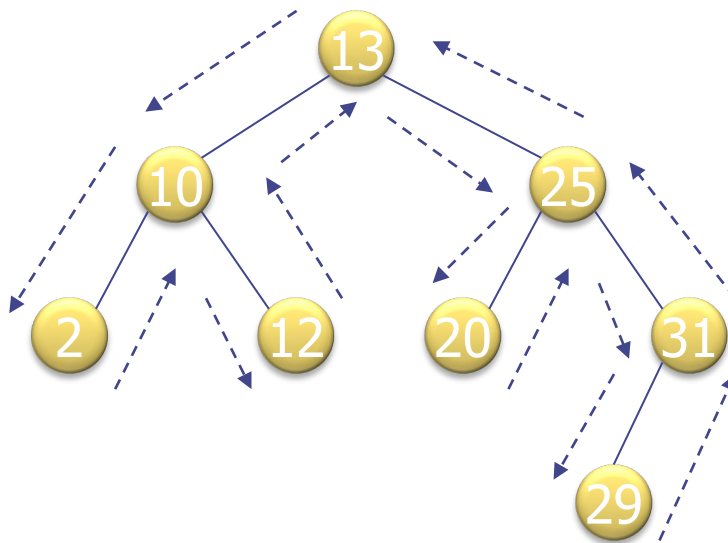
Fila: 31

Fila: 29

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

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)

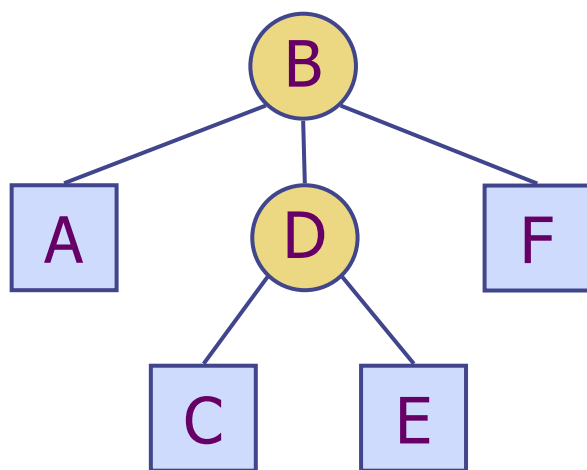


V – Visitar um nó
L – Percorrer à esquerda
R – Percorrer à direita

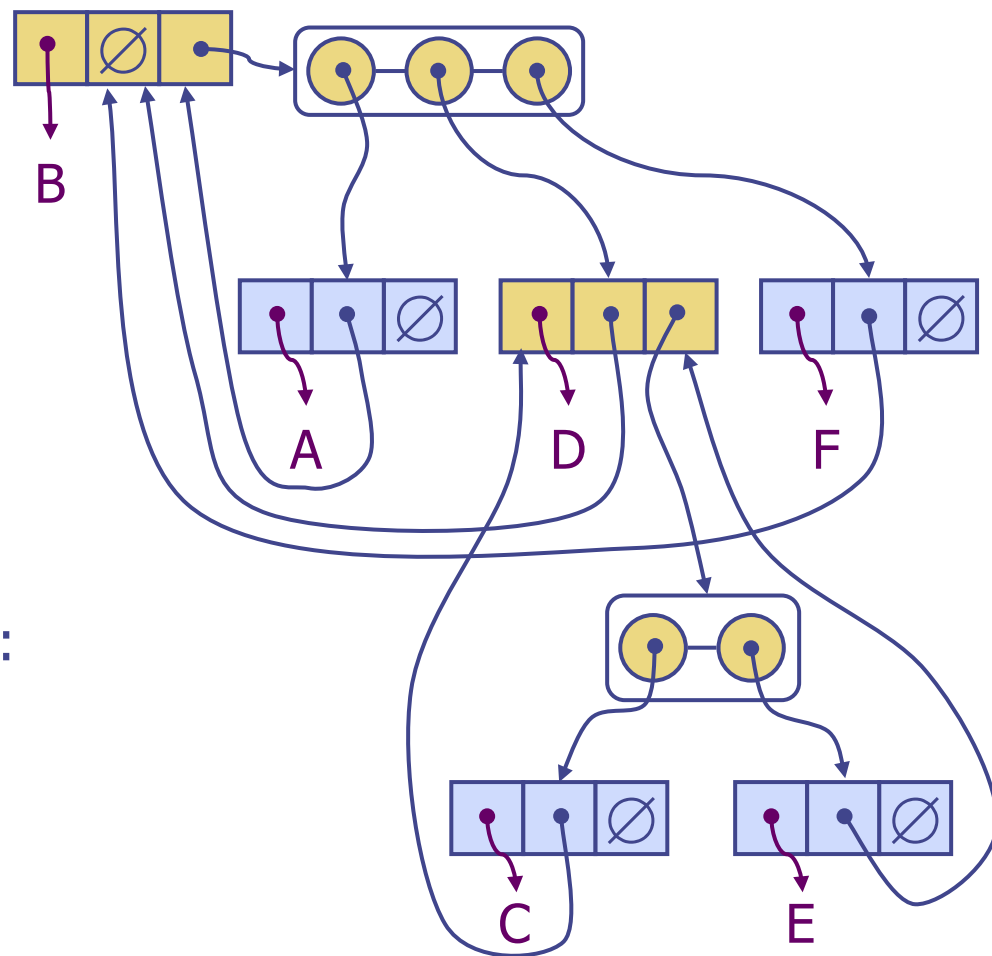
VLR VRL LVR RVL LRV RLV

Um percurso possível: 2, 12, 10, 20, 29, 31, 25, 13

Estrutura de Dados – Árvore Genérica

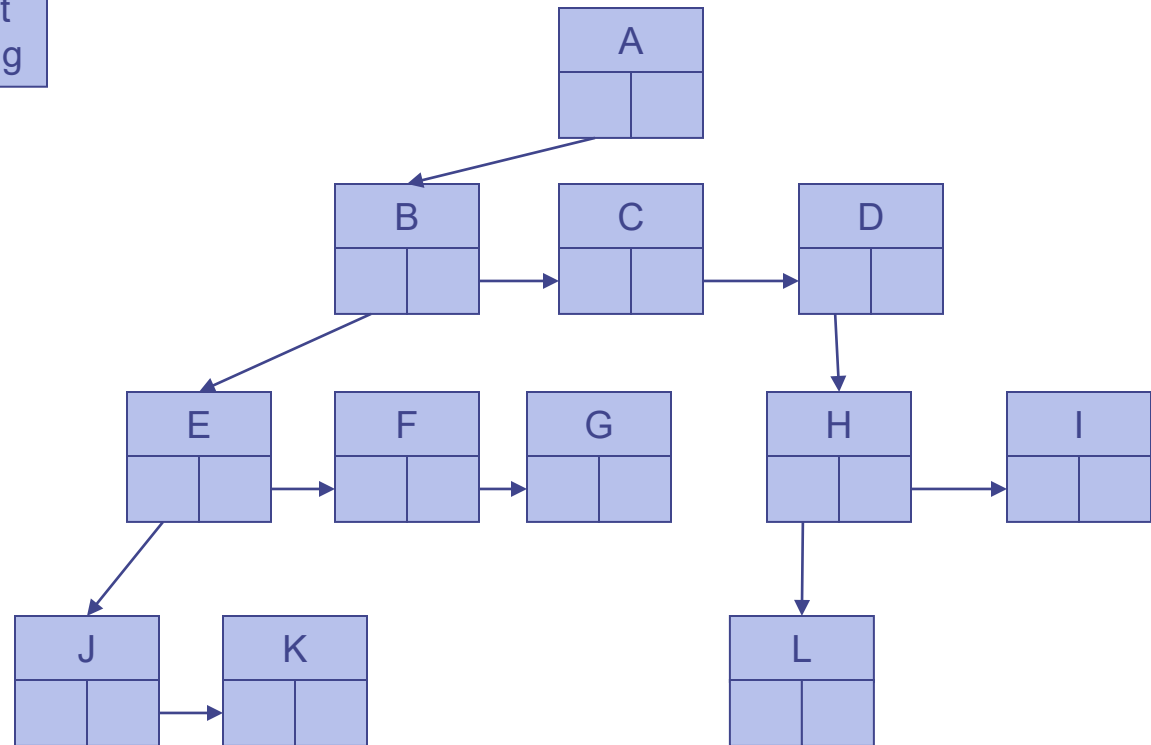


- O nó é representado por:
 - Elemento (dado)
 - Nó pai
 - Lista de filhos



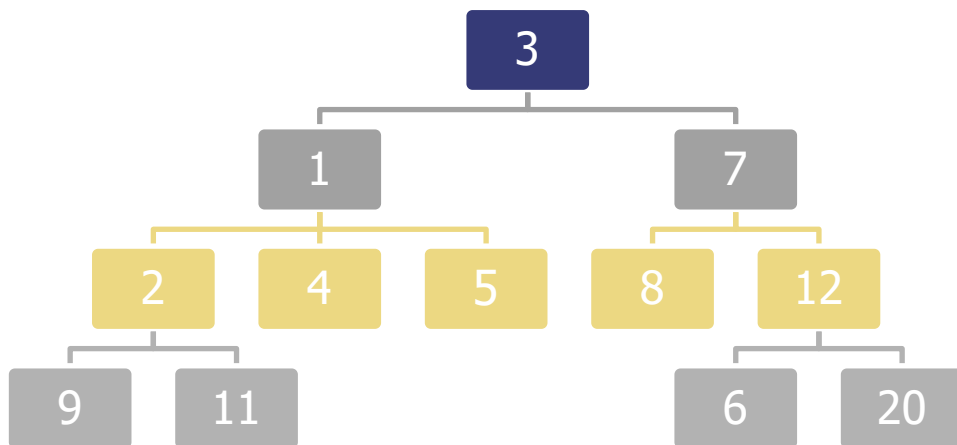
Estrutura de Dados – Representação Filho Esquerdo e Irmão Direito

Data	
Left Child	Right Sibling



Exercícios

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



Número de nós: _____

Nó raiz: _____

Grau do nó 1: _____

Grau da árvore: _____

Número de folhas: _____

Nós internos: _____

Ancestrais do nó 12: _____

Descendentes do nó 7: _____

Irmãos do nó 5: _____

Raiz de uma subárvore: _____

Caminho do nó 3 ao 11: _____

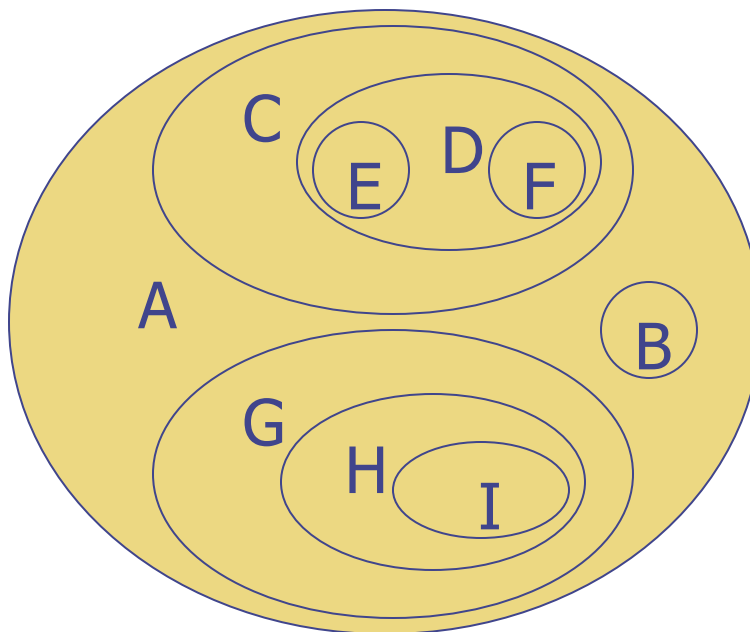
Profundidade do nó 4: _____

Altura do nó 1: _____

Nós do nível 3: _____

Exercícios

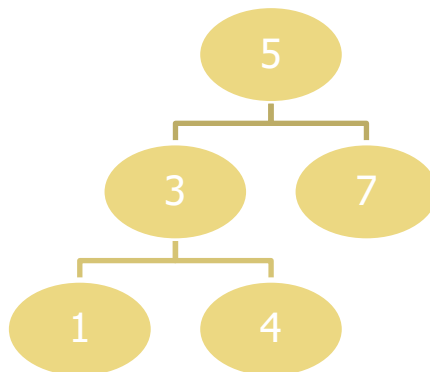
- 2 - Para as representações de árvore abaixo, apresente a árvore em grafo.



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

Exercícios

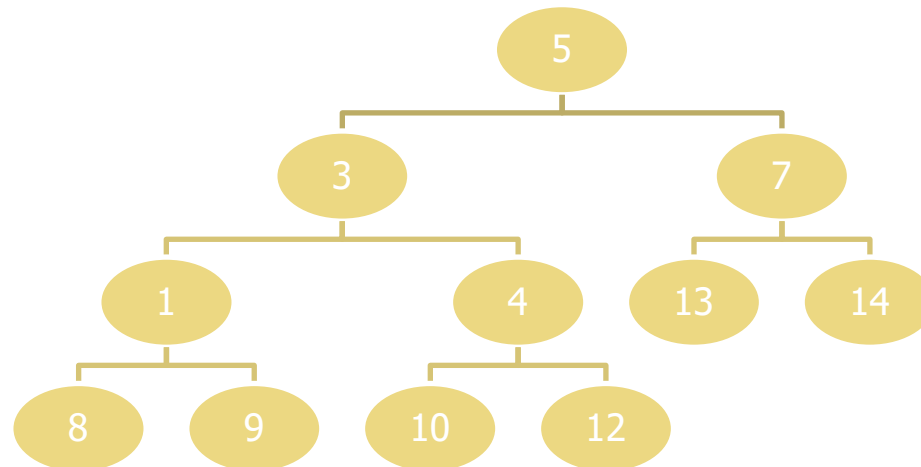
- 3 – Uma das características da utilização de uma árvore é a inserção ordenada (veremos mais adiante em árvores binárias). No lado esquerdo do nó, temos o filho com o menor elemento e, no lado direito do nó, temos o filho com o maior elemento, conforme ilustração abaixo.



- 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 esta seja ordenada: A, B, C, D, E, F, G, 2, 3.**

Exercícios

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



- Apresente o percurso em amplitude;
- Apresente o percurso em profundidade (escolha um).

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.
- **Trees and Binary Trees**. University of North Texas. <http://www.cse.unt.edu/~huangyan>. Acessado em: 28/07/2017.
- **Tree Definitions & Types of Trees**. York University. Gunnar Gotshalks. <http://www.eecs.yorku.ca>. Acessado em: 28/07/2017.