

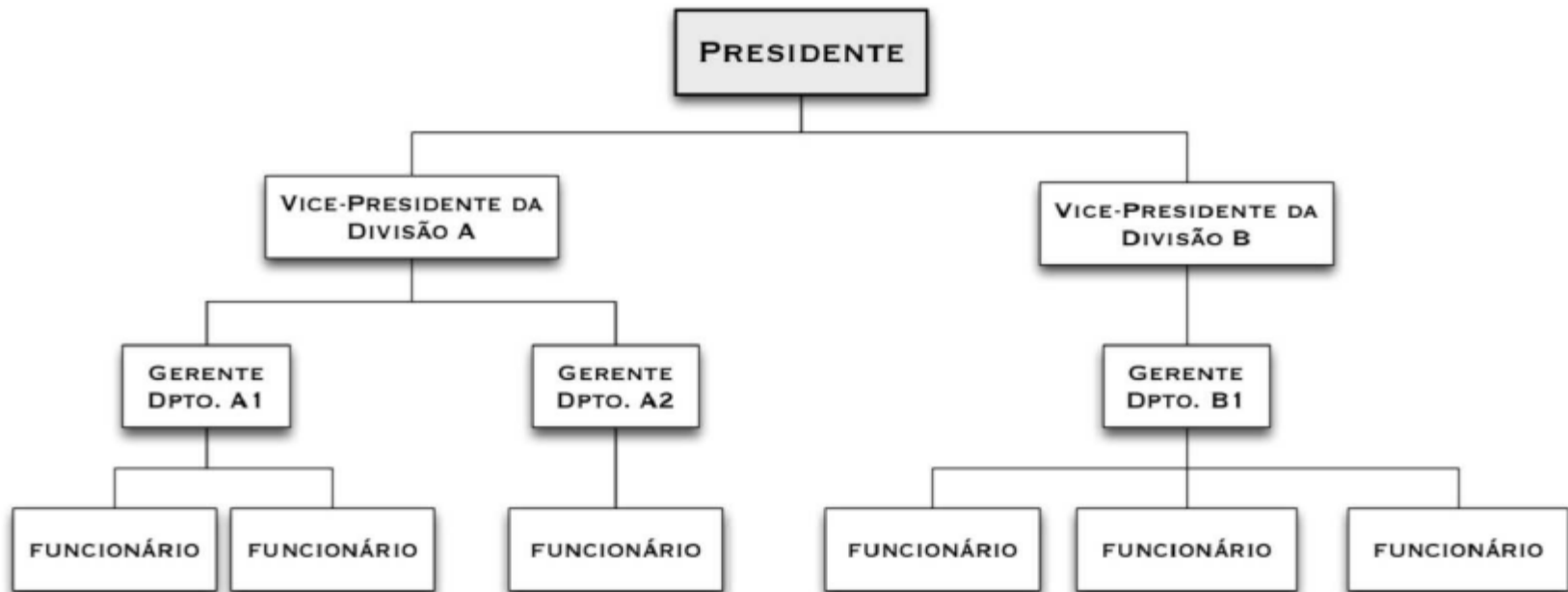
ESTRUTURAS DE DADOS II - ÁRVORES

Prof. Patrícia Noll de Mattos

Árvores: conceito e aplicação

- **Estrutura hierárquica e não linear**
- **Estruturas lineares:** listas encadeadas ou listas sequenciais, possuem normalmente um **elementos anterior e um posterior** a cada nodo.
- As árvores, por serem hierárquicas são representada em **níveis de hierarquia**, conforme a figura a seguir:

Árvore Organograma



- **Presidente** é chamado de RAIZ da árvore e os **funcionários** de FOLHAS.
- Cada **funcionário se reporta a um gerente**, que se reporta a um **vice-presidente**, que se reporta ao **Presidente**.

Árvores: aplicações

- Úteis em diversas aplicações...
- No exemplo anterior, se quisermos saber o **total de salários pagos** aos empregados de uma **divisão ou departamento?**
- Basta **somar os salários** pagos nos **departamentos A1 e A2**, por exemplo, mais o salário do **vice presidente A**.
- Para saber os **salários pagos** no **departamento A1**, basta somar os salários pagos **aos funcionários**, mais o salário pago ao **gerente A1**.

Possível implementação...

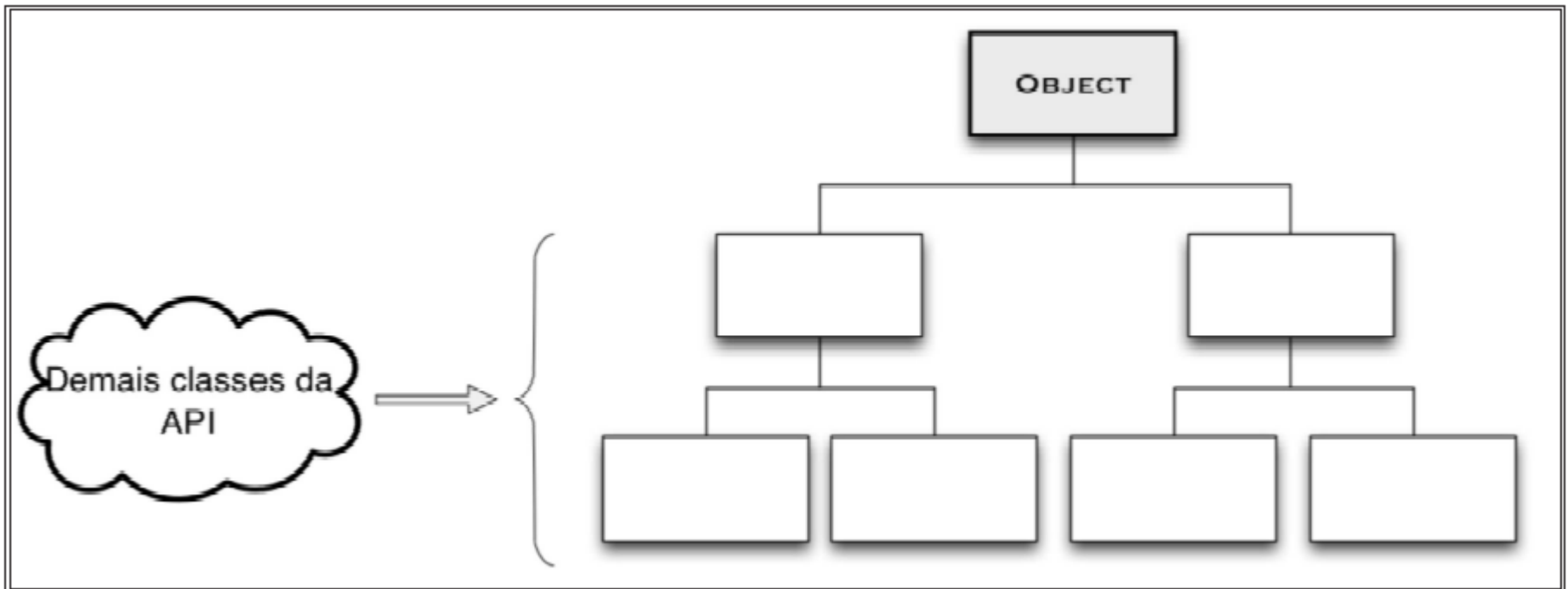
- Visitar sistematicamente cada um dos elementos da árvore de modo a determinar a soma correta dos e salários.
- Um algoritmo que realiza esta visita sistemática é chamado de **caminhamento**.

Terminologias básicas

- Em uma árvore, alguns elementos **estão acima** na **hierarquia** e outros **estão abaixo**.
- Os que **estão acima** são chamados de PAI ou ancestral.
- Os que **estão abaixo** são chamados de FILHOS ou descendentes.
- **Exceto a raiz**, cada elemento da árvore **possui um pai e zero ou mais filhos**.
- Os elementos que possuírem **zero filhos** são chamados de FOLHAS.

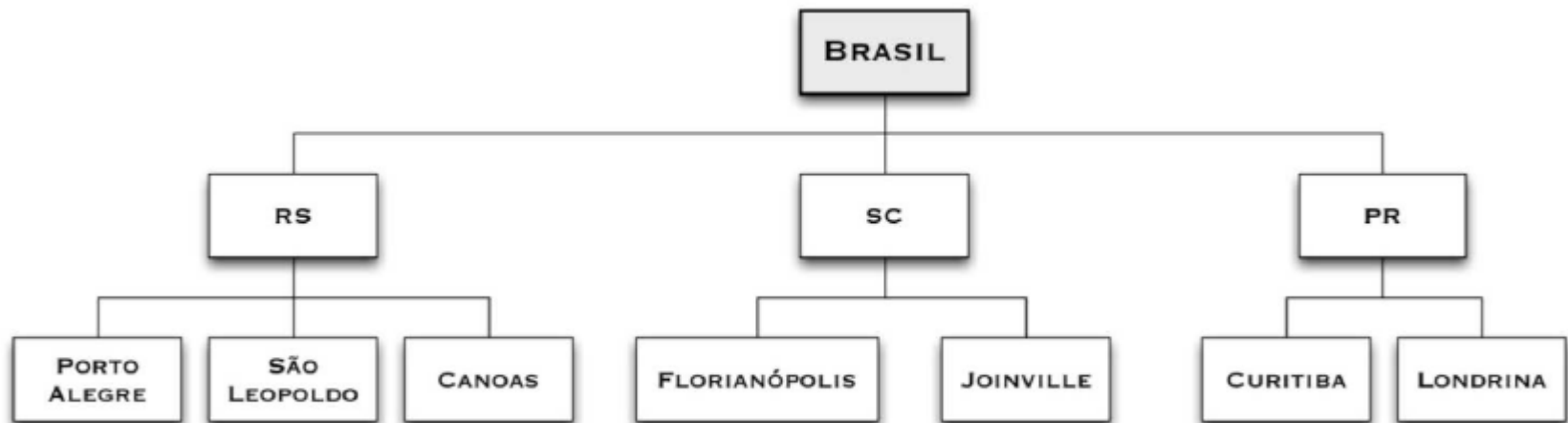
Exemplos de Aplicação

- Hierarquia de classes da **API da linguagem java**.
- Todas as **classes são subclasses** da superclasse Object, diretamente ou indiretamente.



Exemplos de Aplicação

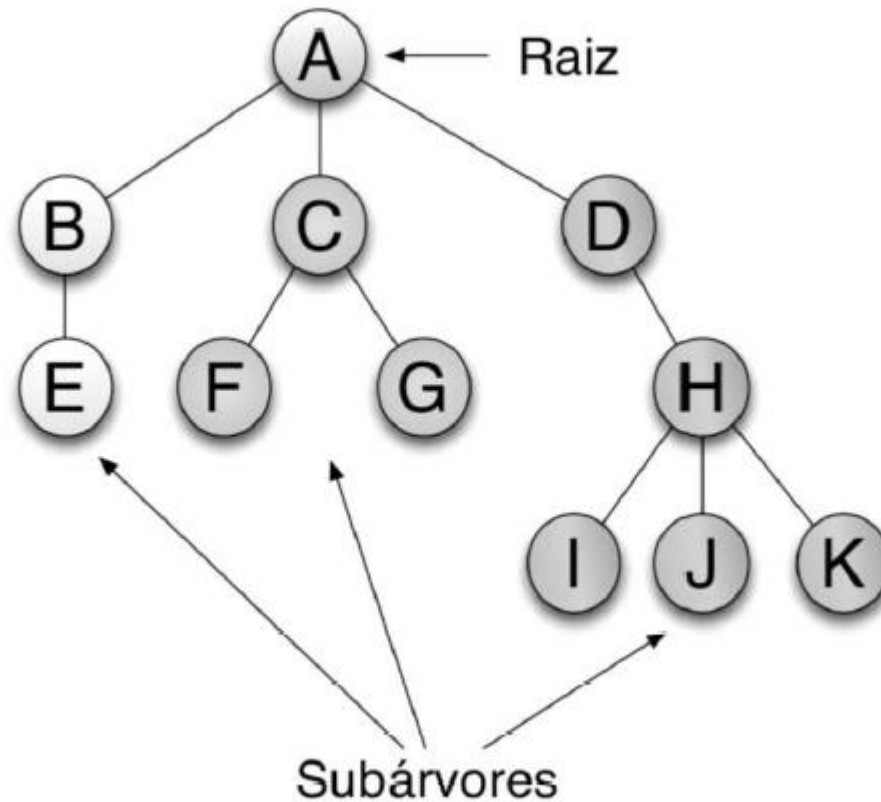
- Representação por **composição**: **país** composto por estados;
- Um **livro** é composto por capítulos.
- Sistemas de arquivos: diretório raiz.
- Representação de um **menu** de opções pode ser representado por uma árvore.



Árvore: definição formal

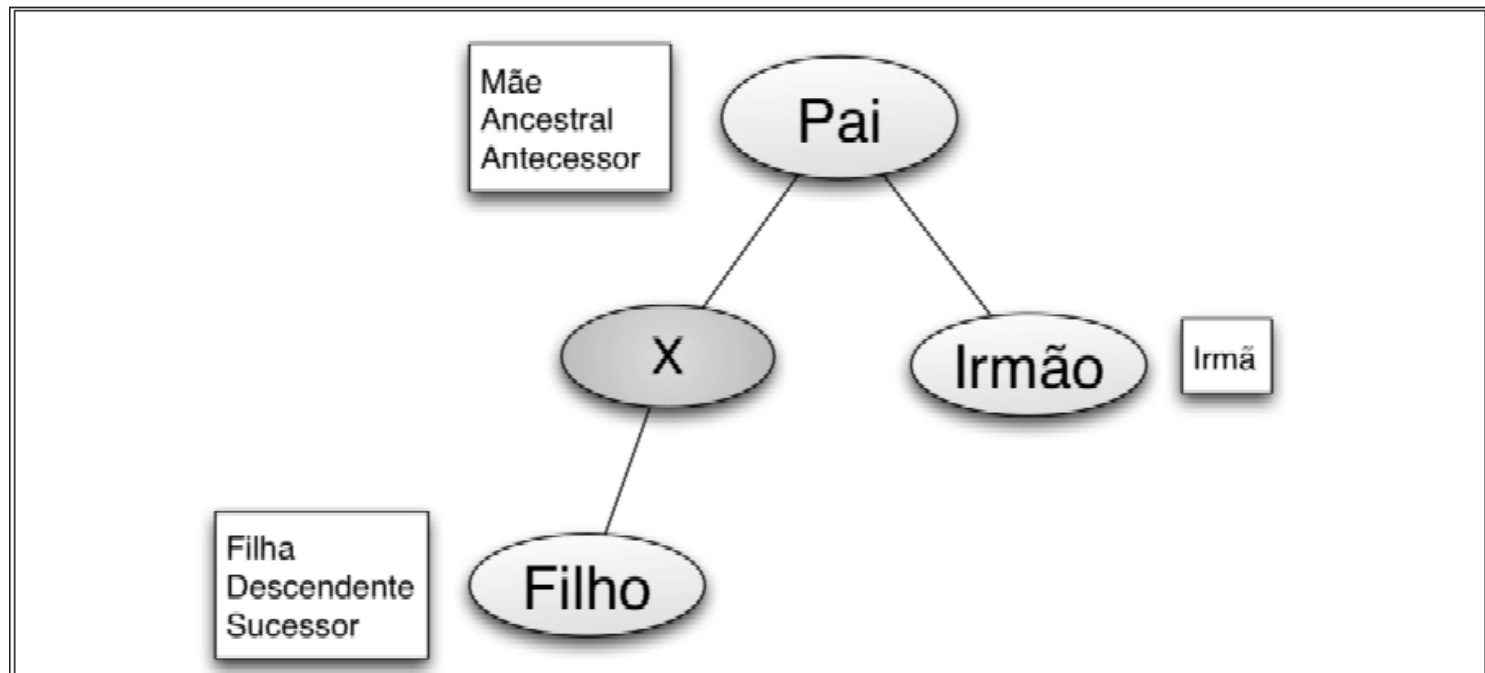
- Uma **árvore T** pode ser definida por um conjunto de **nodos** com relacionamento **pai-filho**:
 - ▣ Se T não for vazia, possui um nodo especial **RAIZ** que **não possui PAI**;
 - ▣ Cada nodo V de T, exceto a raiz, possui um único nodo pai W.
 - ▣ Ou uma árvore está vazia, ou possui um **nodo R**, raiz. R pode ter outras árvores cuja **RAIZ é filha de R**.

Representação por recursão



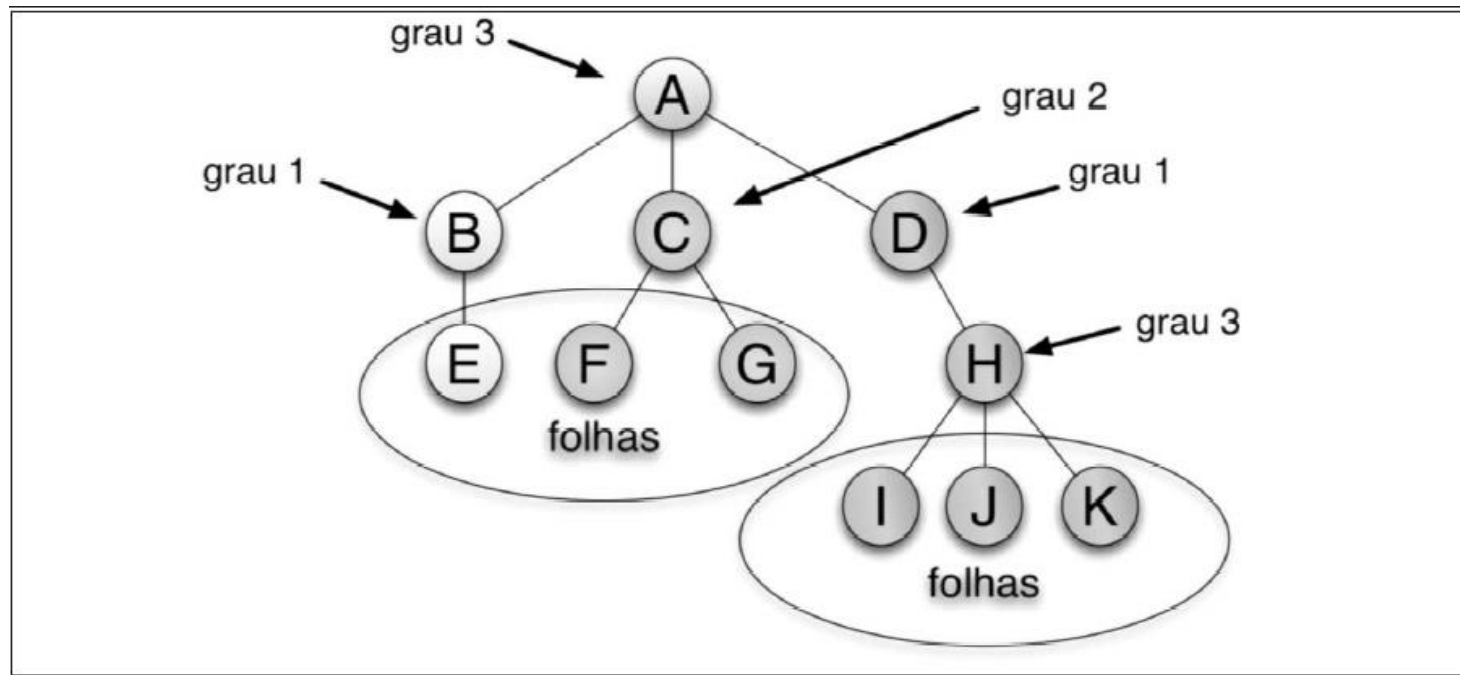
Nodo irmão

- Nodos que possuem o **mesmo pai** são chamados de IRMÃOS.
- **Nodos internos** possuem **filhos**, **nodos externos** não possuem filhos, também chamados de **folhas**.



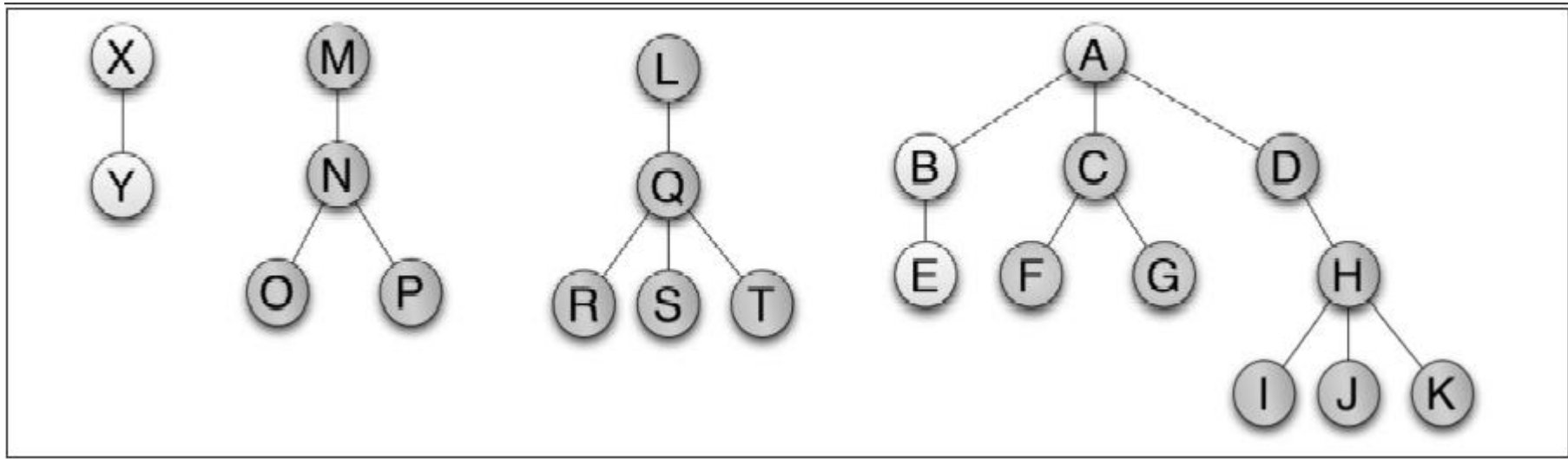
Grau de um nodo

- Determinado pelo **número de subárvores**.
- Nos folha possuem grau zero.
- **Grau de uma árvore: máximo dos graus** de seus nodos.



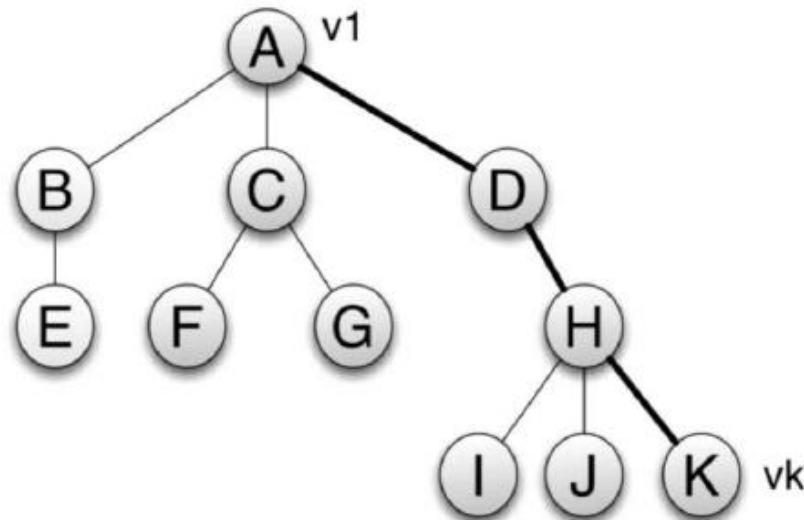
Floresta

- Conjunto de ZERO ou mais árvores disjuntas.



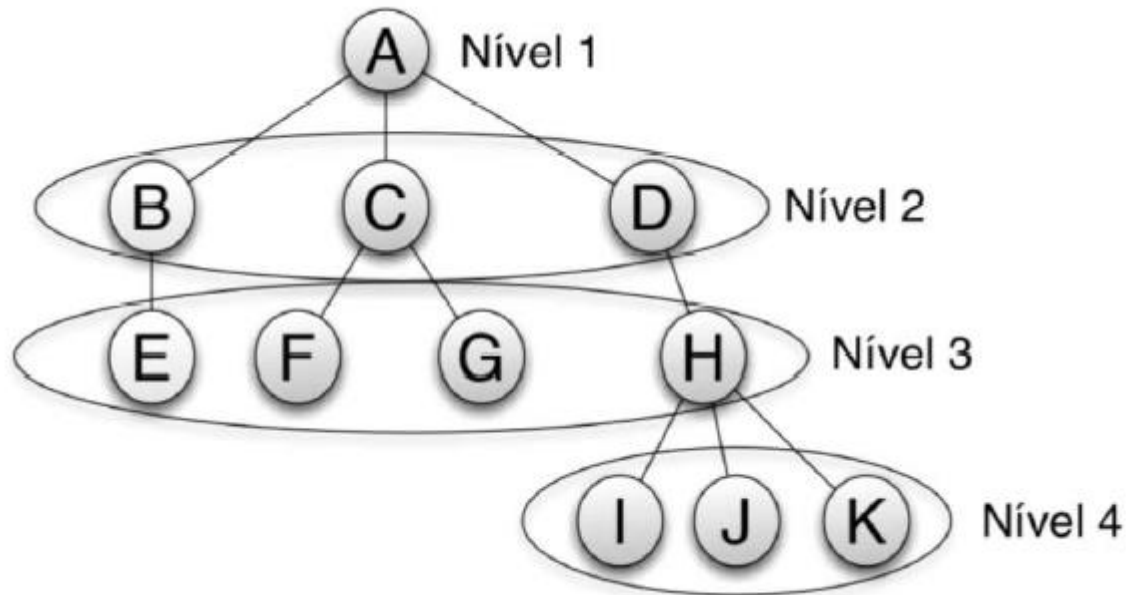
Caminho

- Conjunto de nodos distintos;
- Sempre possuindo a **relação** de “é filho de”, é pai de”.
- **Comprimento de um caminho:** número de nodos do caminho menos 1.



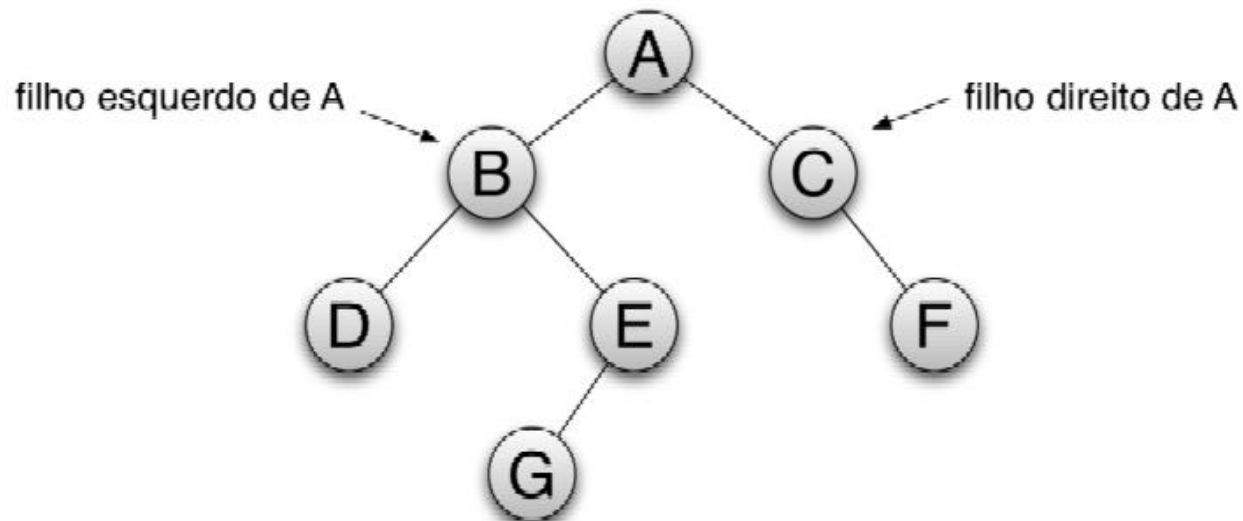
Nível de um nodo

- Número de nodos **entre o nodo avaliado e a raiz**, incluindo-os.
- O nodo raiz tem nível 1.
- A **altura** de uma árvore, ou profundidade é dada pelo **maior nível** entre seus nodos.



Árvores Binárias

- Tipo específico de árvore, onde cada nodo pode conter **zero, um ou dois filhos**.
- Grau máximo é 2.
- Cada nodo: **dados**, **referência ao nodo esquerdo** e **referência ao nodo direito**.

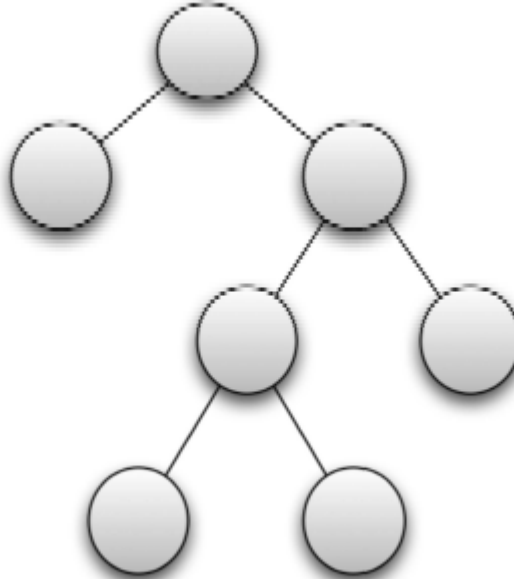


Subárvores

- Nodo **B** é filho **esquerdo de A** e raiz da **subárvore esquerda de A**.
- Nodo **C** é filho **direito de A** e raiz da **subárvore direita de A**.
- Os nodos B e C são **irmãos**.

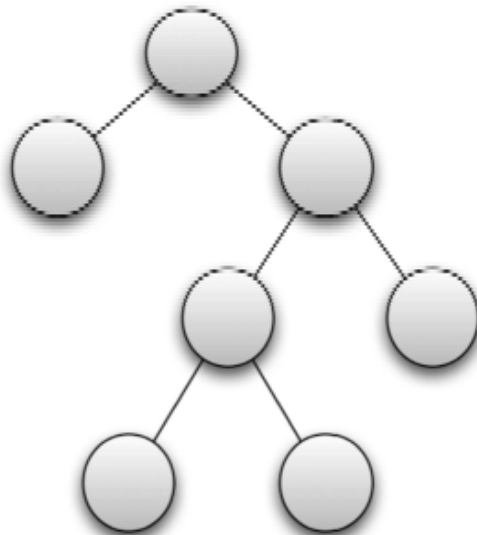
Árvore estritamente binária

- Todo o **nodo interno** da árvore possui subárvores **esquerda** e **direita** não vazias.
- Todos os nodos possuem **zero** (externo) ou **2 filhos**.



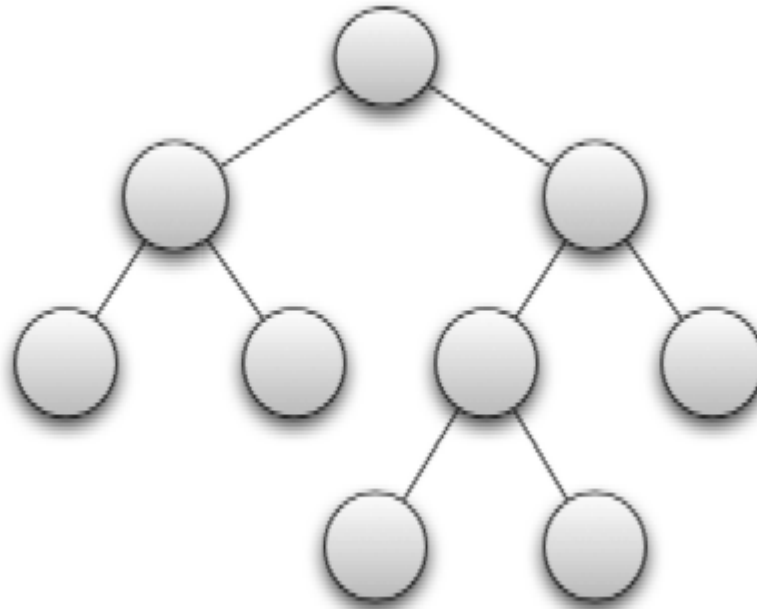
Número de nodos de uma árvore estritamente binária

- $2^n - 1$
- Onde **n** é o número de **nodos folha**
- No exemplo temos 4 nodos folha:
 - $2^4 - 1 = 7$



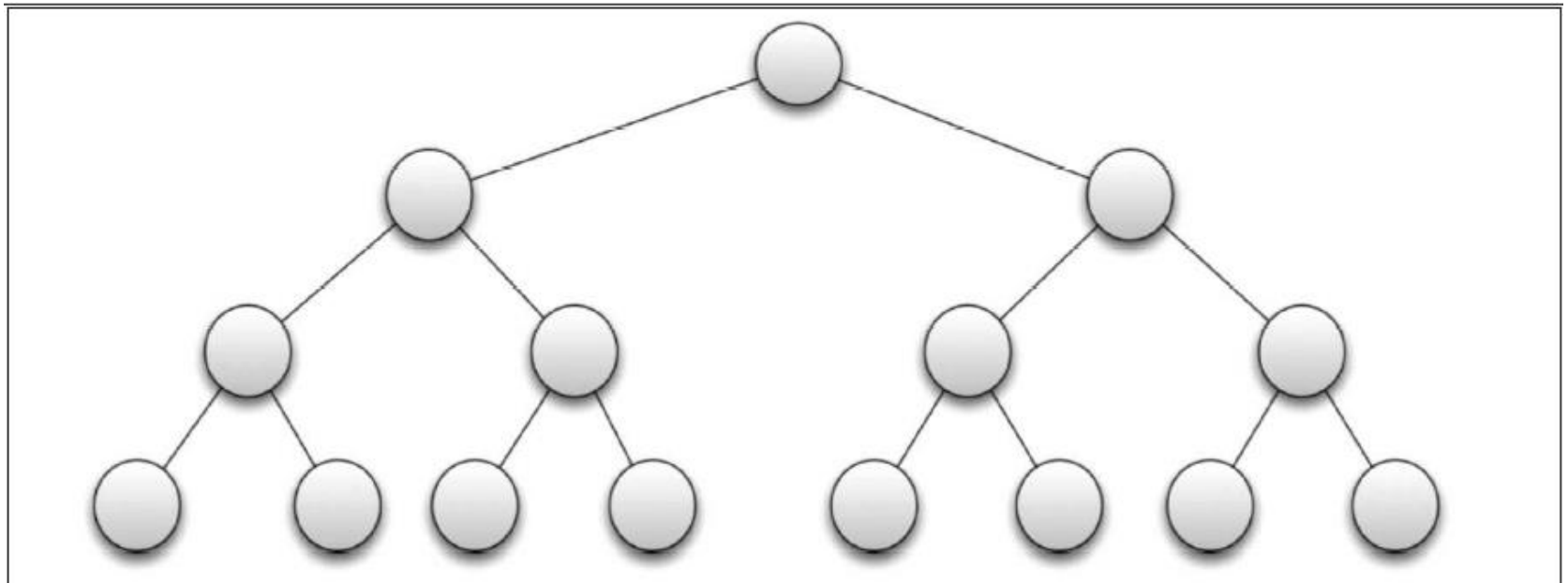
Árvore Binária Completa

- Todos os nodos **folha** encontram-se no **último** ou penúltimo **nível**.



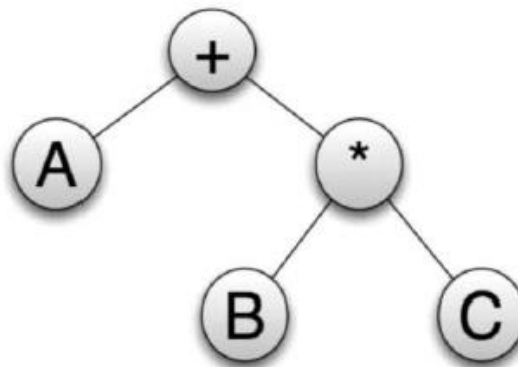
Árvore Binária Completa Cheia

- Todas as folhas encontram-se no **último nível**.
- **Número total de nodos** para uma árvore de **altura h** é $2^h - 1$.
- Para o exemplo temos $2^4 - 1 = 16 - 1 = 15$.



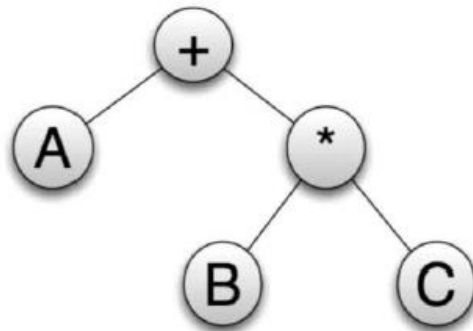
Aplicação de Árvores Binárias

- Exemplo: representação de uma **expressão aritmética**.
- Cada **nodo interno da árvore** representa um **operador**;
- Cada **nodo externo** da árvore representa um **operando** (constante ou variável).
- É sempre uma árvore estritamente binária.
- Operador com **menor prioridade** aparece sempre na **raiz** da árvore.



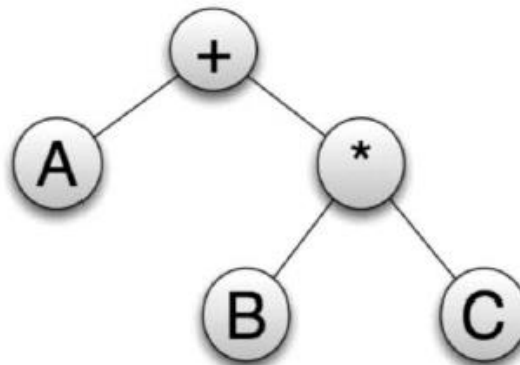
Prioridade das Operações

- A **subexpressão à esquerda** deste operador dá origem à **subárvore esquerda**.
- A **subexpressão à direita** deste operador dá origem à **subárvore direita**.
- Ordem de prioridade das operações fica implícita.
- **Caminhamento** da árvore: **avalia a expressão**



Caminhamentos

- **Pós ordem:** notação pós-fixada: $A B C * +$
 - ▣ Sub esq – sub dir - nodo
- **Em ordem:** notação infixa (normal): $A+B*C$
 - ▣ Sub esq – nodo – sub dir
- **Pré ordem:** notação polonesa: $+A*BC$
 - ▣ Nodo – sub esq – sub dir
- Podem ser utilizados para reescrever a expressão.



Árvore Aritmética – Pseudocódigo

- Suponha uma árvore representando um expressão aritmética contendo apenas operadores: $+$, $-$, $*$ e $/$

Interpretar(String expressão) {

1. Construir uma árvore com a expressão, contendo um único nodo;
2. Procurar por operador de menor precedência ($+$ ou $-$), que não esteja entre parênteses. Se encontrado, substituir a árvore com este operador como raiz e operandos como nodos folha. Repetir o processo recursivamente no nodo direito da árvore.
3. Repetir o passo 2 nos operadores de média precedência ($*$ ou $/$), apenas nos nodos folha.
4. Percorrer os nodos folha (apenas operandos) : Se tiverem números constantes ou variáveis, nada a fazer; Se for uma expressão entre parênteses, substituir este nodo por Interpretar(expressão sem parênteses),

}

Passo 1:

- Nodo é criado contendo a expressão inteira:

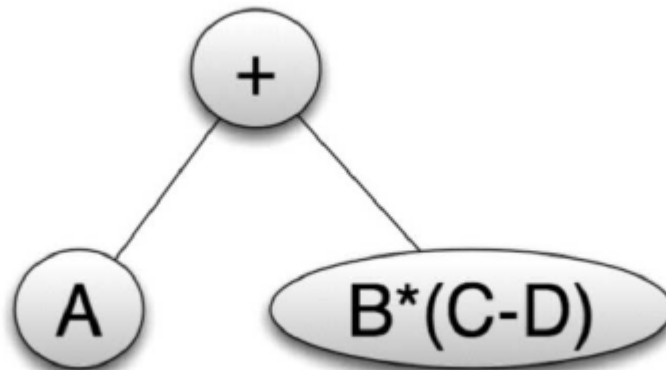


A diagram showing a node. It consists of a light gray oval with a black border, containing the text $A+B*(C-D)$. This oval is centered within a larger white rectangle that has a thin black border.

$$A+B*(C-D)$$

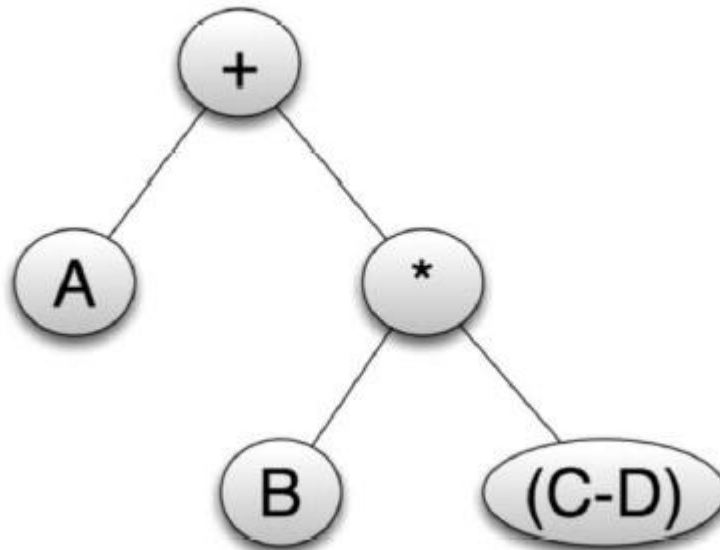
Passo 2

- Analisar a expressão da **esq para dir** para encontrar o **operador de menor precedência**.
- Operador é a **nova raiz** da árvore.



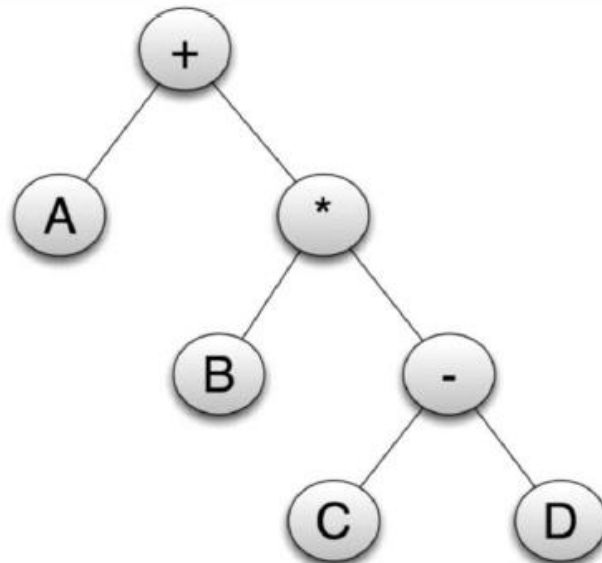
Passo 3

- Não existem outros **operadores de menor precedência** (parênteses são operandos)
- Procura de **média precedência**:



Passo 4:

- ❑ Não existem mais **operadores de média** precedência.
- ❑ A única expressão é (C-D)
- ❑ Chama o algoritmo recursivamente:



Algoritmo para avaliar a expressão

Suponha a existência de uma definição de “registro arvoreexpressao” que represente um nodo da árvore, contendo os campos esquerdo, direito, operando e operador.

```
real avalia(registro arvoreexpressao *v) {  
    real retorno  
    se (v == nulo) retorno = 0  
    senao se (v->operador == ' ') retorno = v->operando  
    senao se (v->operador == '+') retorno = avalia(v->esquerda) + avalia(v->direita)  
    senao se (v->operador == '-') retorno = avalia(v->esquerda) - avalia(v->direita)  
    senao se (v->operador == '*') retorno = avalia(v->esquerda) * avalia(v->direita)  
    senao se (v->operador == '/') retorno = avalia(v->esquerda) / avalia(v->direita)  
    retorna retorno  
}
```

Definição de cada nodo

registro arvoreexpressao

inicio

char operador

real operando

registro arvorebinaria *esquerda

registro arvorebinaria *direita

fim

registro arvoreexpressao *raiz=nulo

Exercício:

- Apresente passo a passo a construção da árvore binária para a expressão a seguir:
 - ▣ $A + B - C / (C + A * B)$
 - ▣ Apresente o teste de mesa de avaliação desta expressão.