

Introdução a Árvores

Regina Célia Coelho

rccoelho@unifesp.br

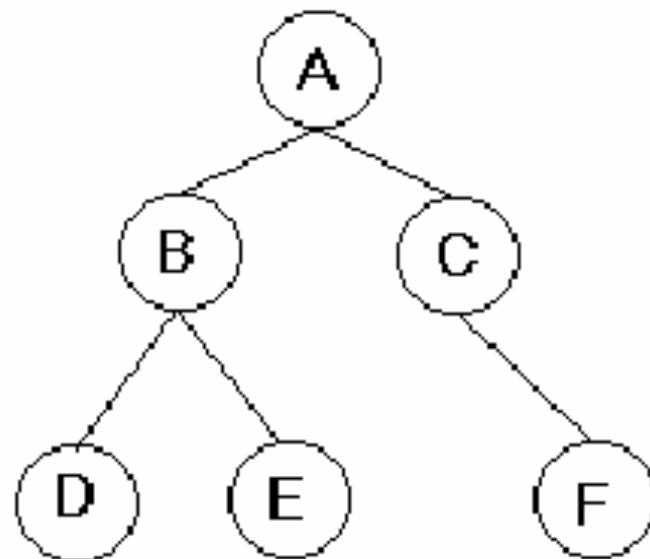
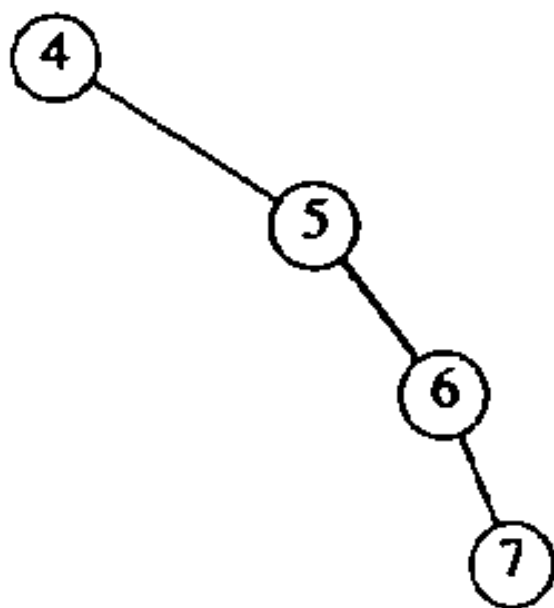
Introdução

- Uma das mais importantes classes de estruturas de dados em computação são as árvores.
- Aproveitando-se de sua organização hierárquica, muitas aplicações são realizadas usando-se algoritmos relativamente simples, recursivos e de eficiência bastante razoável.

- Uma árvore é uma estrutura de dados que se caracteriza por uma relação de hierarquia entre os elementos que a compõem.
- Exemplos de estruturas em forma de árvores
 - ✓ O organograma de uma empresa;
 - ✓ A divisão de um livro em capítulos, seções, tópicos, etc;
 - ✓ A árvore genealógica de uma pessoa.
- De um modo um pouco mais formal, podemos dizer que uma árvore é um conjunto finito de um ou mais nós (vértices), tais que:
 - ✓ existe um nó denominado raiz;
 - ✓ os demais nós formam $m \geq 0$ conjuntos separados s_1, s_2, \dots, s_m , tais que cada um desses conjuntos também é uma árvore (denominada **sub-árvore**).

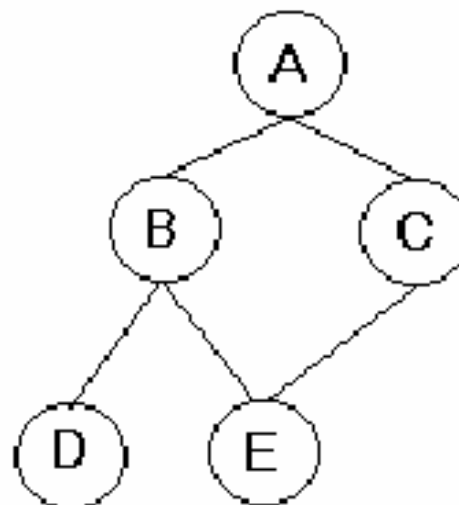
Representação

- Uma lista é um tipo de árvore degenerada, pois cresce somente para um lado.
- Representação gráfica utilizando grafos:



Representação (cont.)

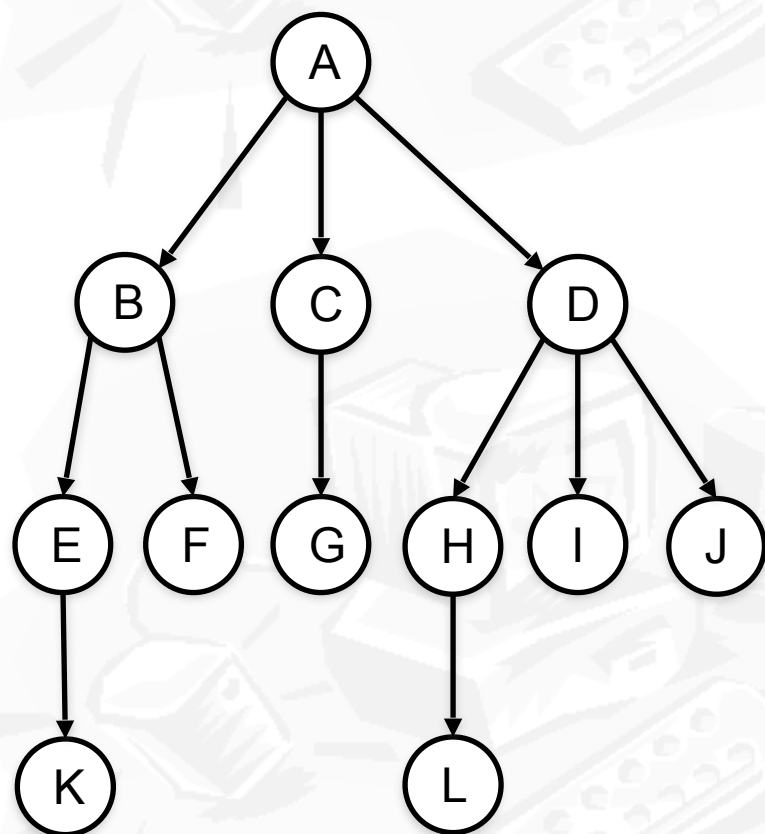
- Uma árvore é um grafo conexo acíclico.
- Cada nó só pode ter um pai.
- Assim, o desenho abaixo, por exemplo, não representa uma árvore.



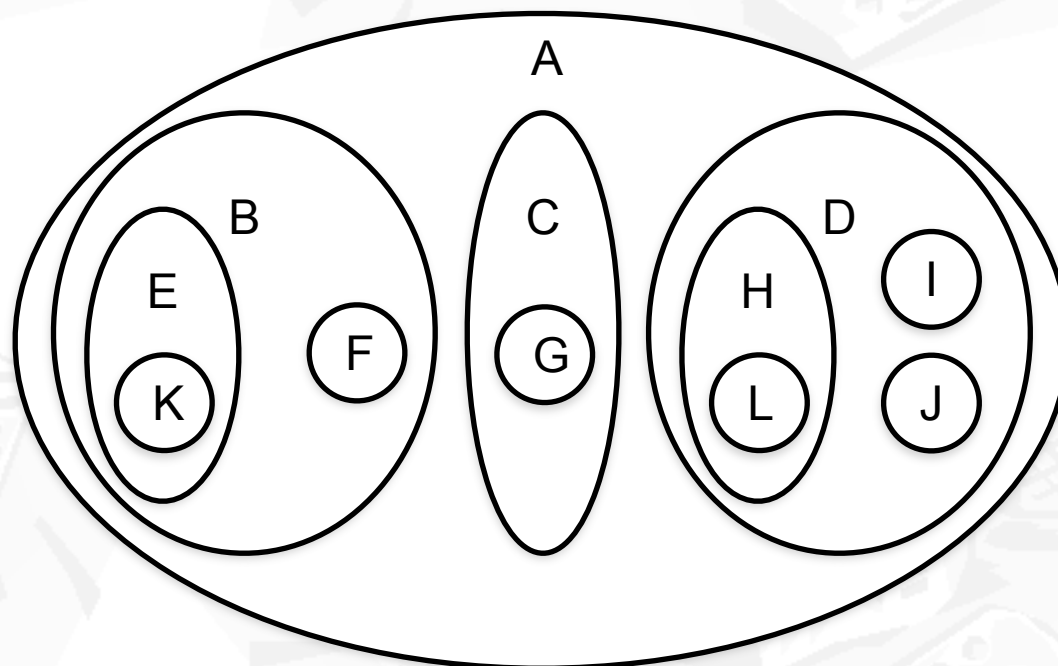
Representação (cont.)

➤ Diferentes representações de árvores:

➤ **Grafo**



➤ **Conjuntos aninhados**



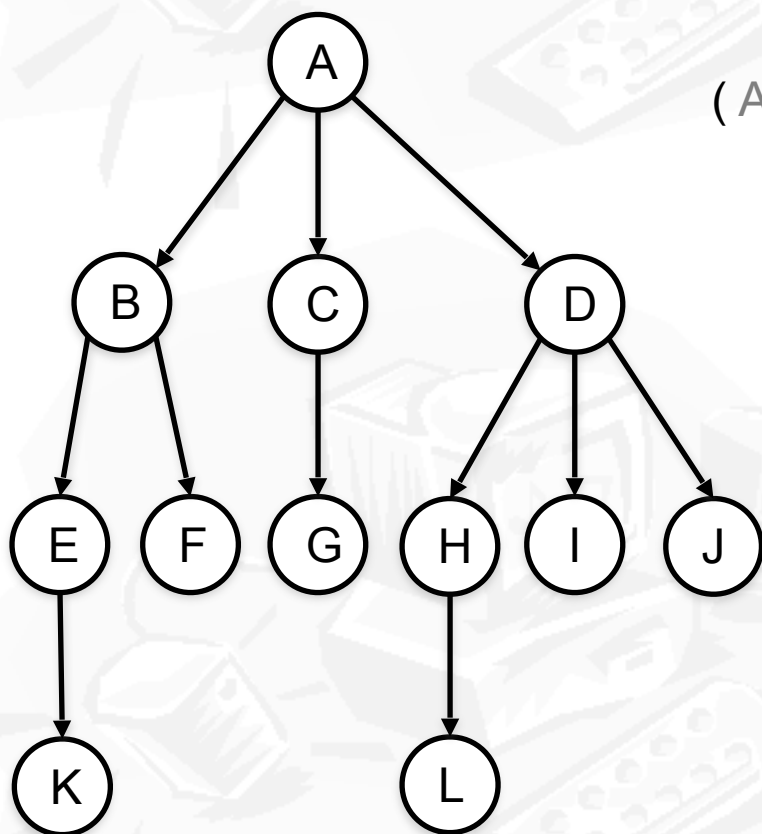
Representação (cont.)

➤ Diferentes representações de árvores:

➤ **Grafo**

➤ **Parênteses aninhados**

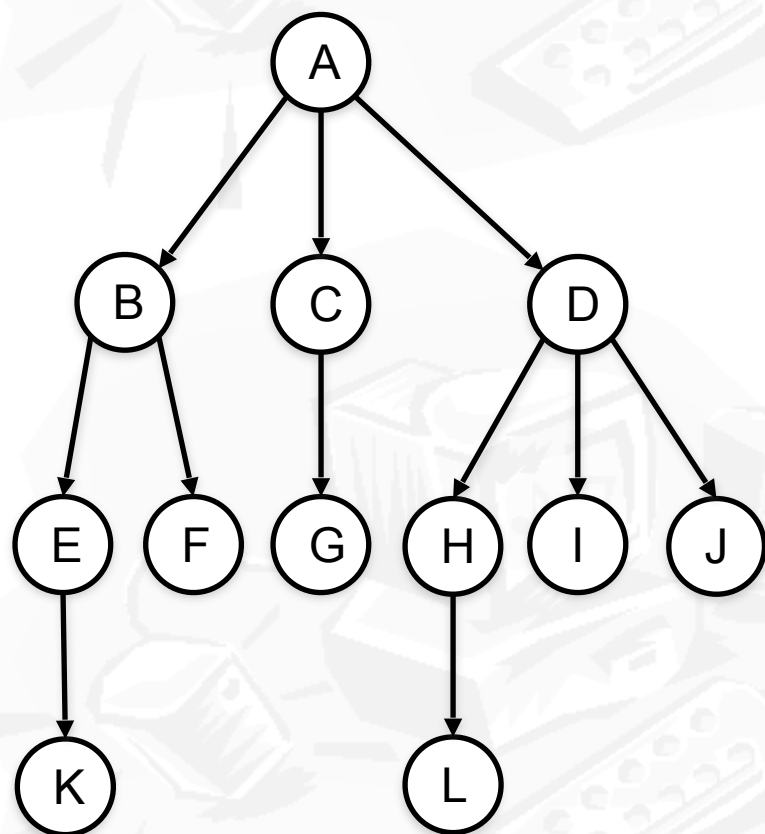
(A(B(E(K)F)C(G)D(H(L)IJ)))



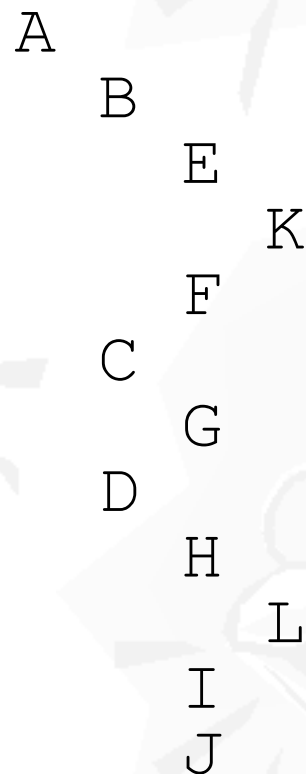
Representação (cont.)

➤ Diferentes representações de árvores:

➤ Grafo



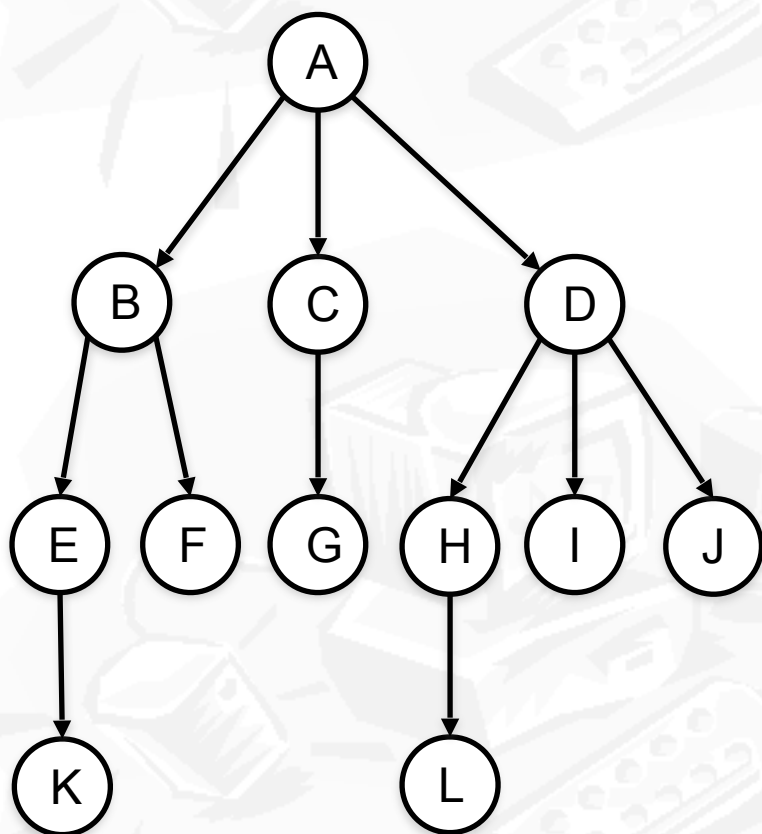
➤ Paragrafação



Representação (cont.)

➤ Diferentes representações de árvores:

➤ Grafo

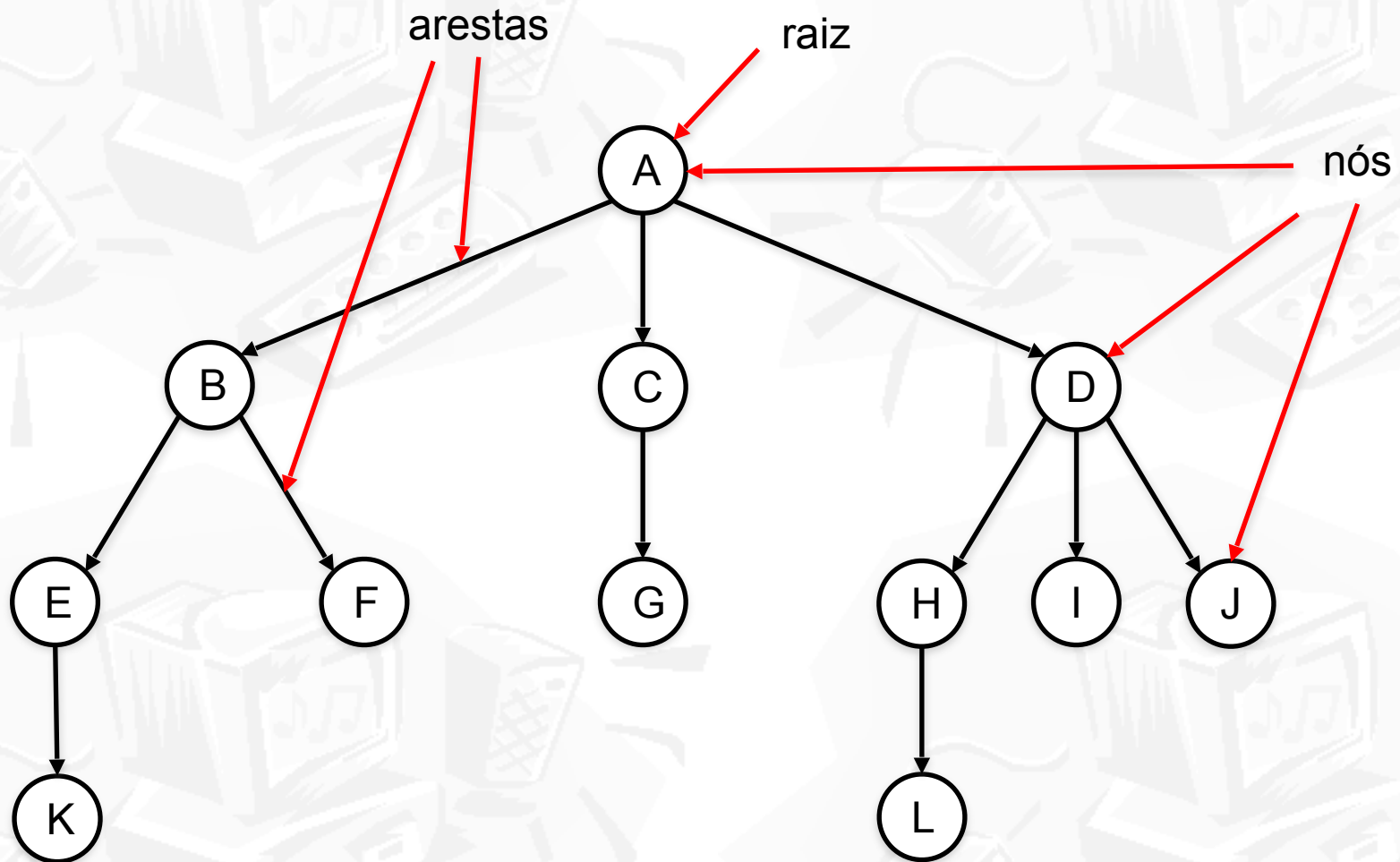


➤ Paragrafação

A
 . . B
 E
 K
 F
 . . C
 G
 . . D
 H
 L
 I
 J

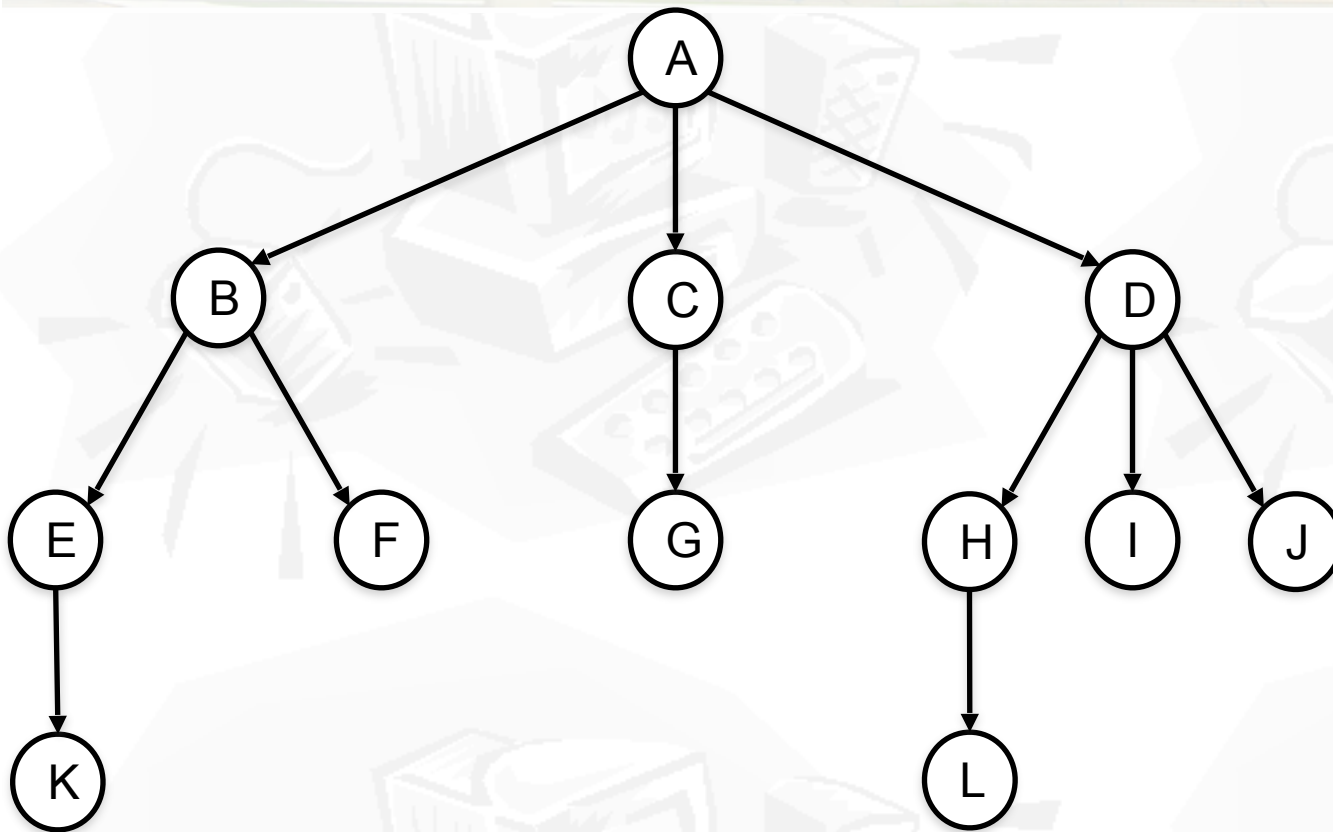
- Dada uma árvore qualquer, a linha que liga dois nós da árvore denomina-se **aresta**.
- Diz-se que existe caminho entre dois nós V e W da árvore, se a partir do nó V for possível chegar ao nó W percorrendo-se as arestas que ligam os nós intermediários entre V e W .
- Observa-se que existe sempre um caminho entre a raiz e qualquer nó da árvore.
- Se houver um caminho entre V e W , começando em V diz-se que V é um nó ancestral de W e W é um nó descendente de V .
 - ✓ Se este caminho contiver uma única aresta, diz-se que V é o nó **pai** de W e que W é um nó **filho** de V .
 - ✓ Dois nós que são nós filhos do mesmo nó pai são denominados **nós irmãos**.

Conceitos básicos (cont.)



- Normalmente as árvores são desenhadas de forma invertida, com a raiz em cima.

Conceitos básicos (cont.)

**Irmãos**

B, C, D

E, F

H, I, J

Nó Pai

A

B

C

D

E

F

G

H

I

J

K

L

Nós Filhos

B, C, D

E, F

G

H, I, J

K

-

-

L

-

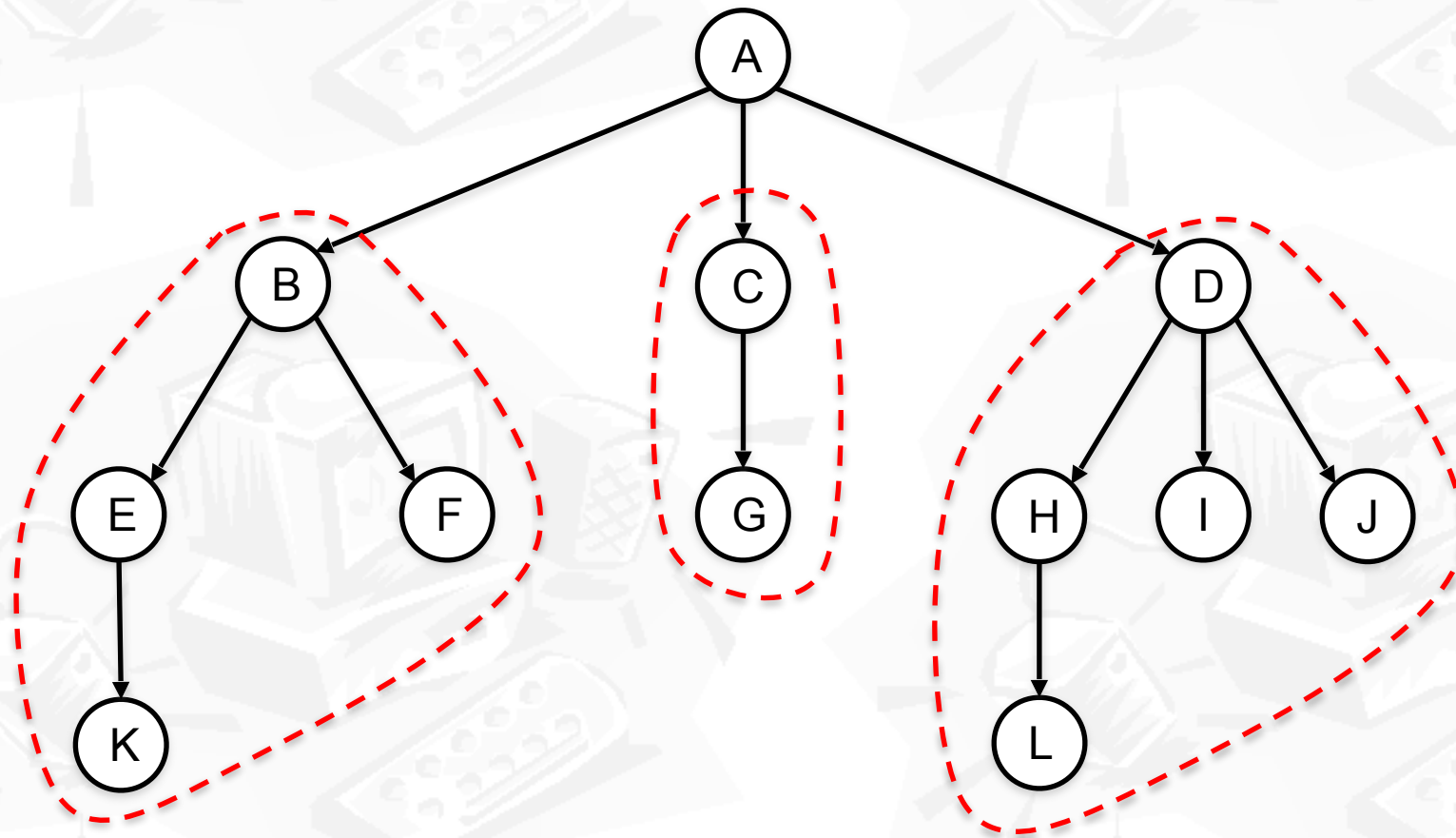
-

-

-

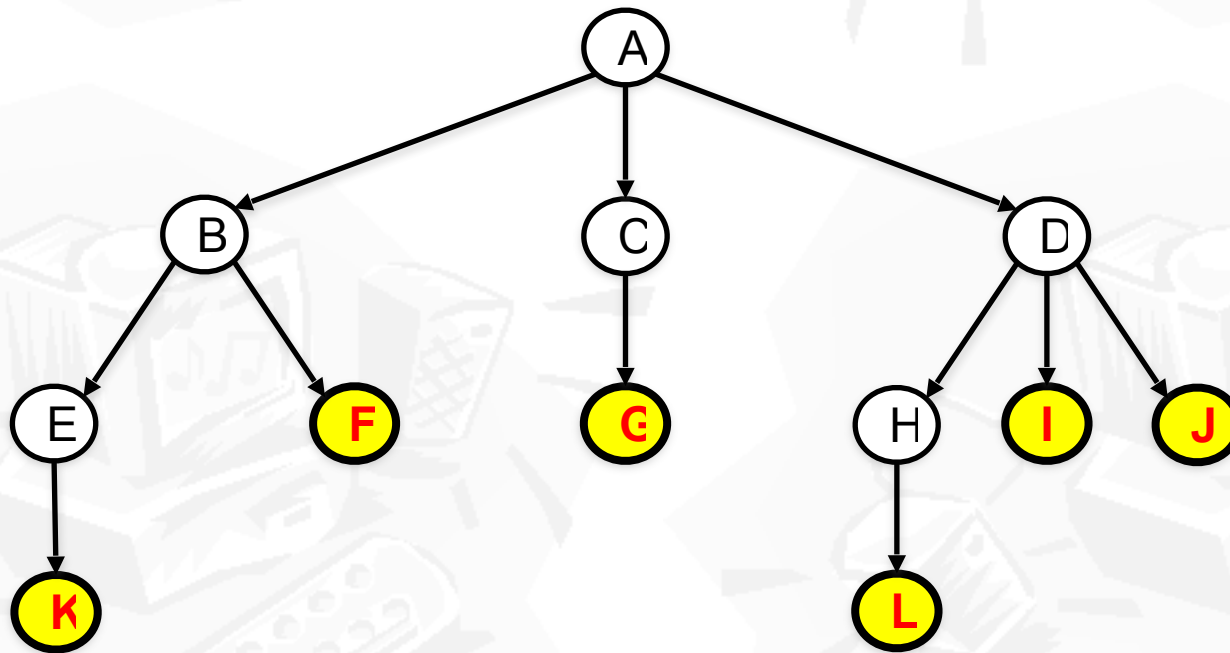
Conceitos básicos (cont.)

- No exemplo, o nó A possui três **subárvores** (ramos) cujas raízes são B, C e D



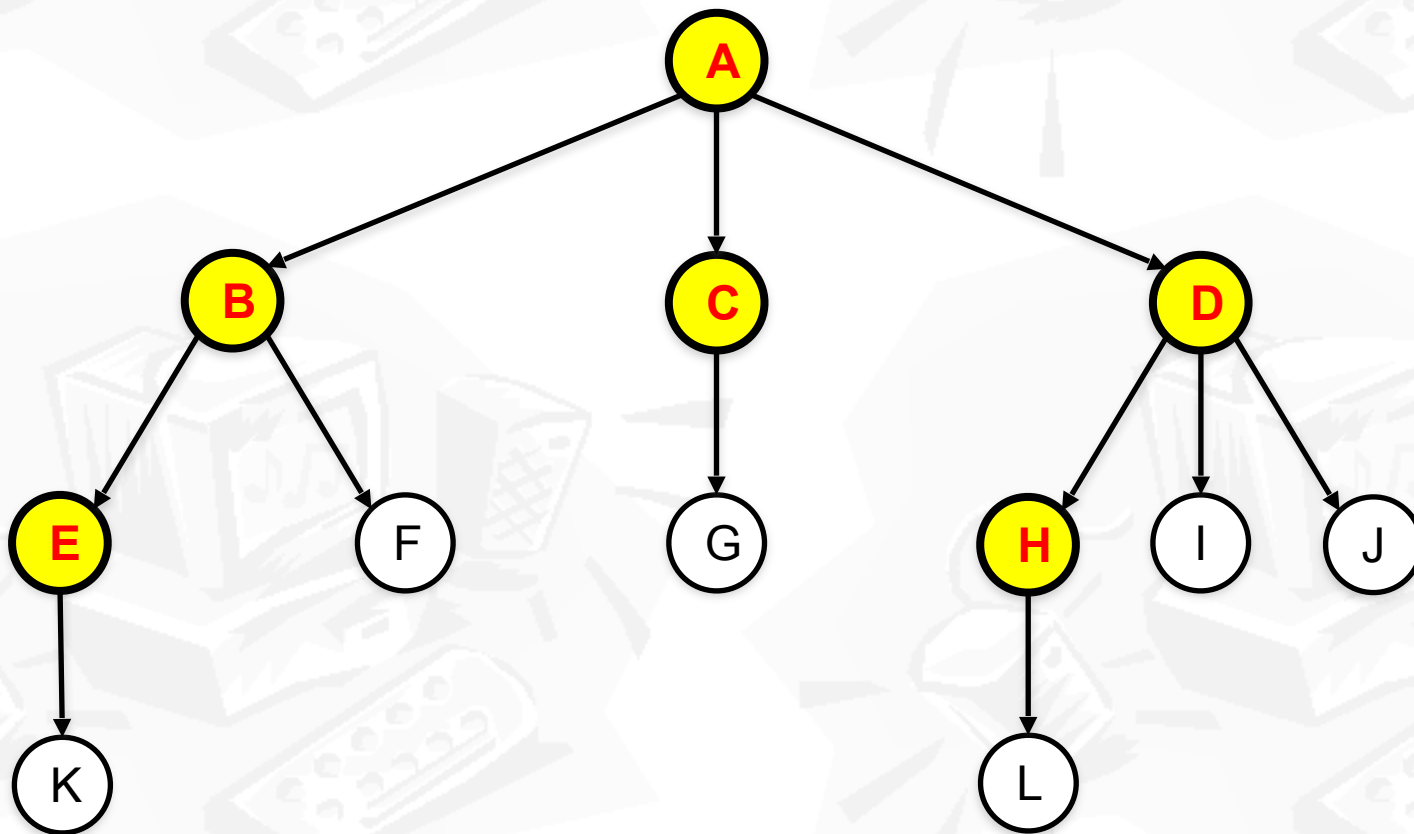
Conceitos básicos (cont.)

- Uma característica inerente a árvores é que qualquer nó, exceto a raiz, tem um único nó pai.
- Se um nó não possui nós descendentes, ele é chamado de **folha** ou **nó terminal** da árvore (nós com grau zero).



Conceitos básicos (cont.)

- Consequentemente, nó com descendentes (filhos) é denominado nó **não-folha** ou **nó não-terminal** da árvore (nós com grau maior que zero).



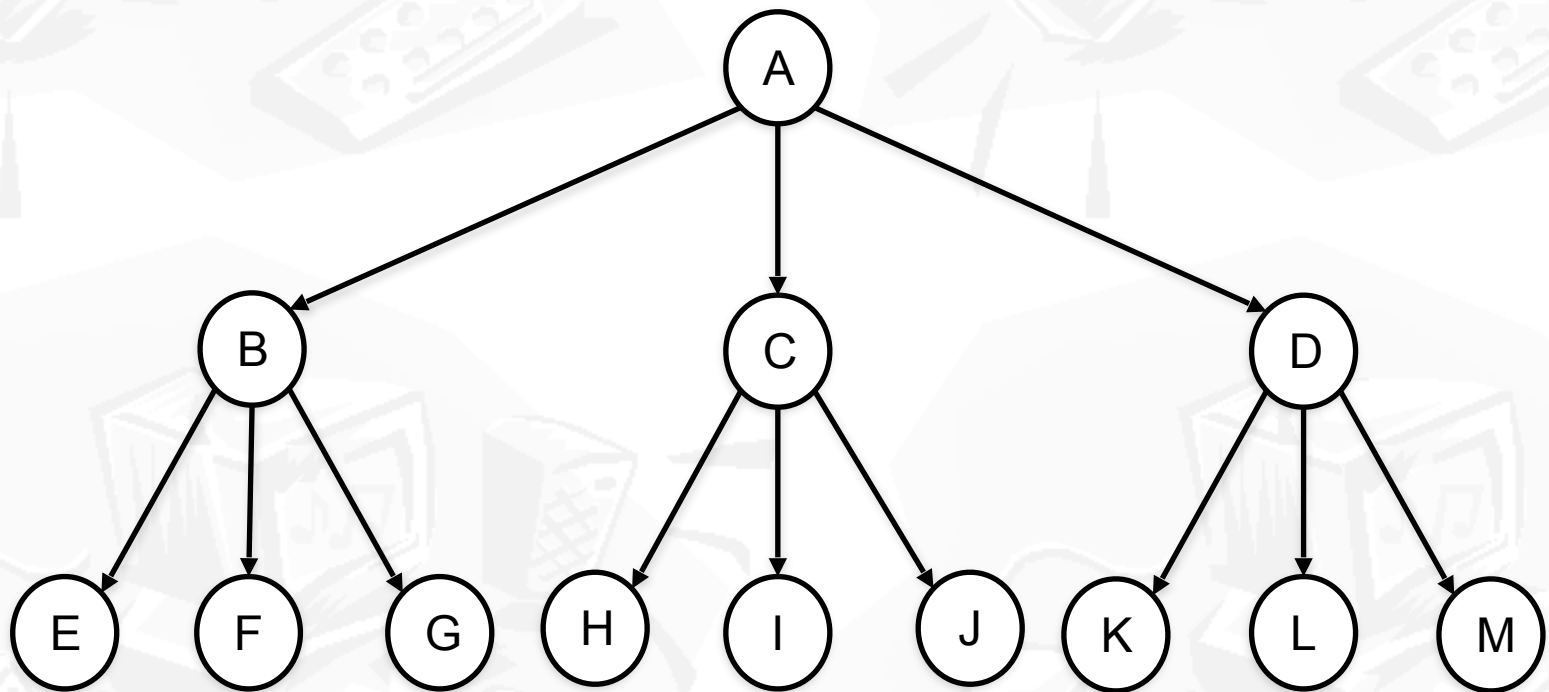
Conceitos básicos (cont.)

- **Grau** de um nó é o número de nós filhos do mesmo nó. Obviamente que um nó folha tem grau zero.
- **Nível** de um nó é o número de nós existentes no caminho entre a raiz e o próprio nó.
 - ✓ Por definição, dizemos que a raiz de uma árvore encontra-se no nível 0 (alguns autores assumem a raiz como nível 1).
 - ✓ Os filhos da raiz estão no nível 1, os filhos dos filhos da raiz estão no nível 2, e assim sucessivamente.
 - ✓ Estando um nó no nível n , seus filhos estarão no nível $n + 1$.

Conceitos básicos (cont.)

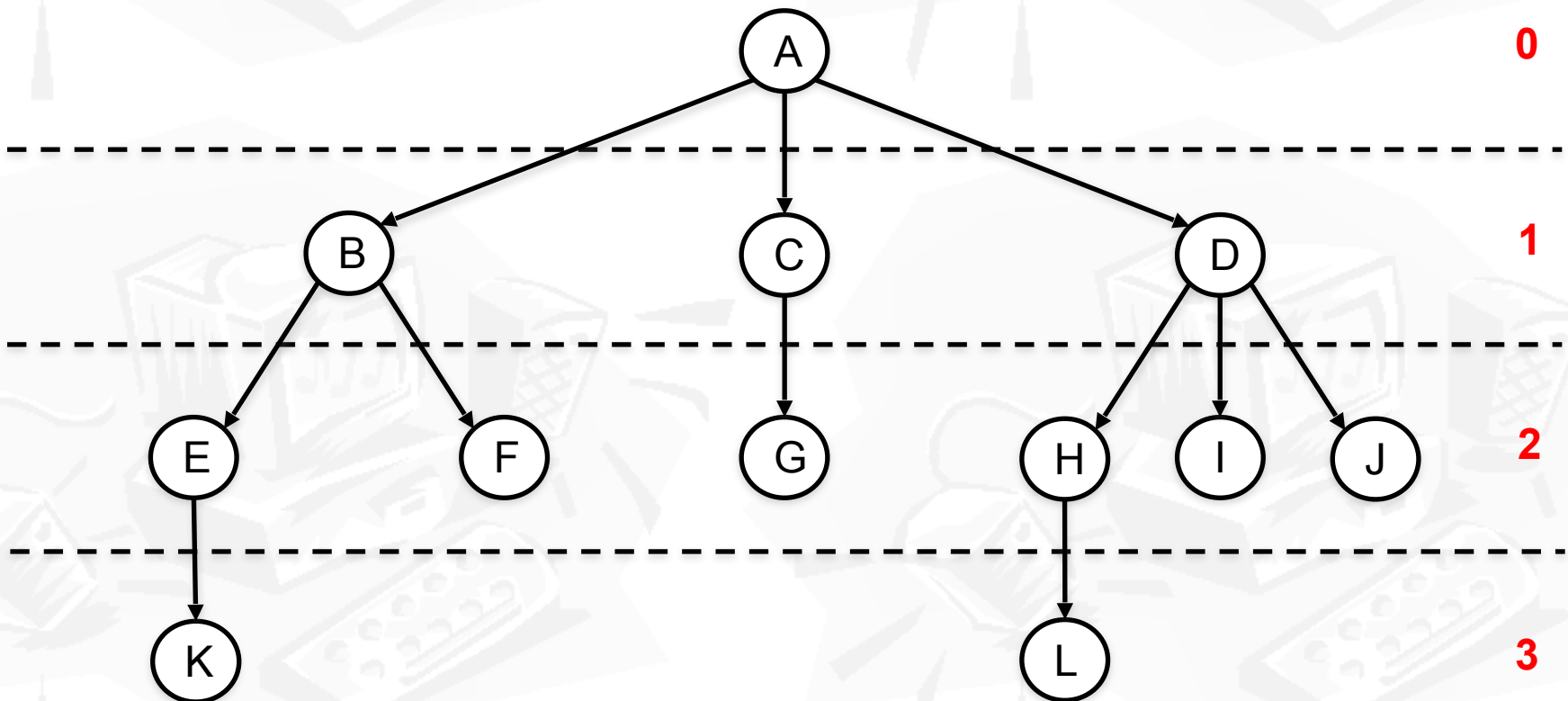
- O grau da árvore é igual ao grau do nó de maior grau da árvore.
- Uma árvore de grau **d** é **completa** se:
 - ✓ Todos os nós têm exatamente **d** filhos, exceto as folhas;
 - ✓ Todas as folhas estão no mesmo nível.
- O **nível** da árvore é igual ao nível do nó de maior nível da árvore.
- Um conjunto de n árvores separadas ($n \geq 0$) é chamado de **floresta**.
 - ✓ Se retirarmos a raiz de uma árvore obteremos uma floresta.

➤ Exemplo de árvore completa



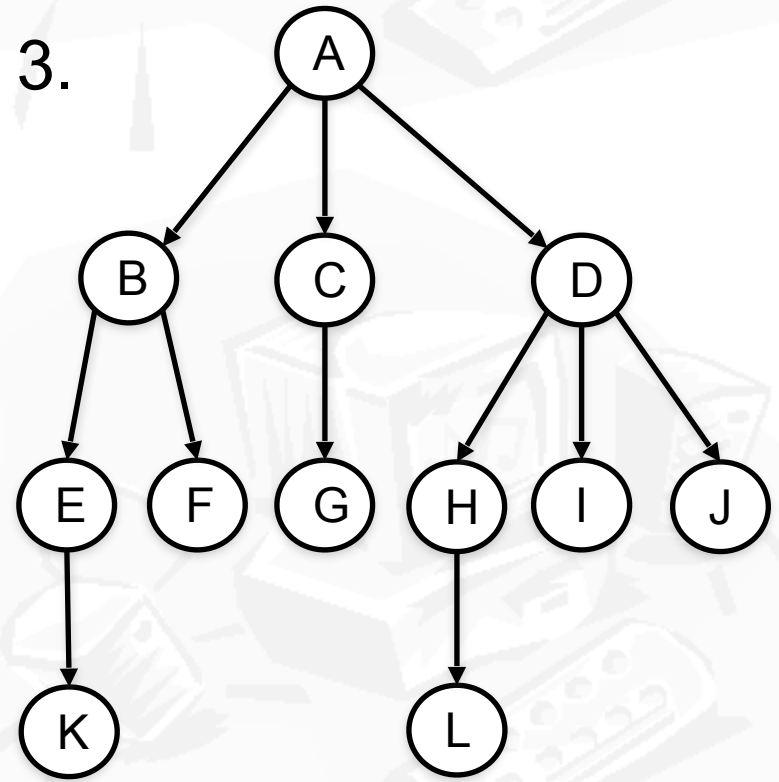
Conceitos básicos (cont.)

- B, C e D estão no nível 1
- K e L estão no nível 3



Conceitos básicos (cont.)

- A **altura** (ou **profundidade**) de uma árvore é o nível máximo entre todos os nós da árvore ou, equivalentemente, é a altura da raiz.
- No exemplo, a árvore possui altura 3.
- A altura de uma árvore vazia é -1.



Conceitos básicos (cont.)

- O número máximo de nós $n(h, d)$ em um árvore de altura h é atingido quando todos os nós possuírem d subárvores, exceto o nível h , que não possui subárvores.
- Para uma árvore de grau d :
 - ✓ Nível 0 contém d^0 (um) nó (raiz)
 - ✓ Nível 1 contém d^1 descendentes da raiz
 - ✓ Nível 2 contém d^2 descendentes
 - ✓ ...
 - ✓ Nível i contém d^i descendentes

Conceitos básicos (cont.)

➤ Assumindo **d** = 3

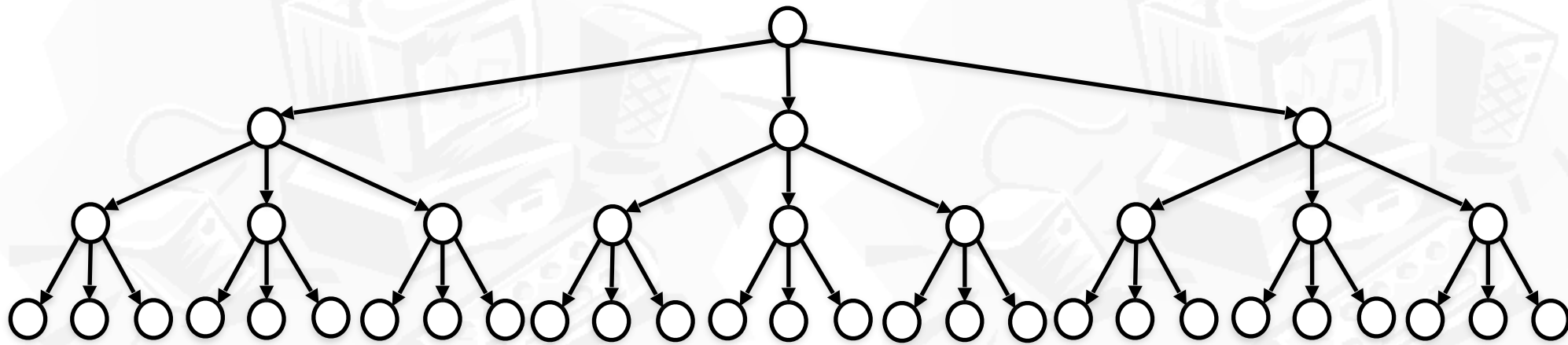
✓ Nível 0: 1 nó (raiz)

✓ Nível 1: 3 nós

✓ Nível 2: $3^2 = 9$ nós

✓ Nível 3: $3^3 = 27$ nós

➤ $n(3, 3) = 1 + 3 + 9 + 27 = 40$ nós



Conceitos básicos (cont.)

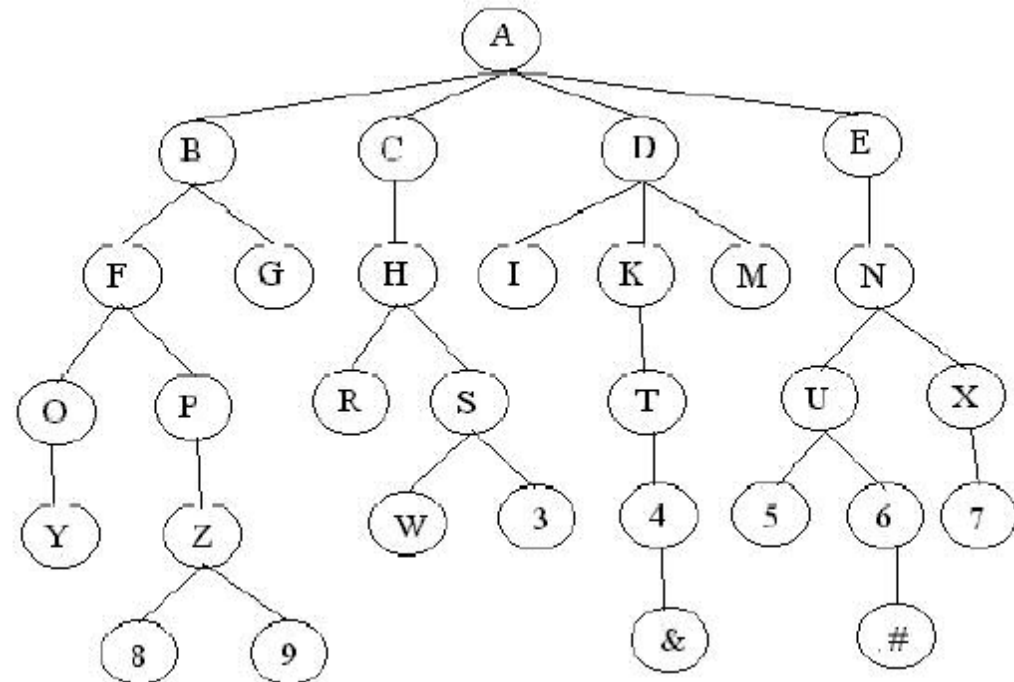
- Portanto, o número máximo de nós $n = n(h, d)$ é a soma do número de nós em cada nível, ou seja:

$$n = n(h, d) = \sum_{i=0}^h d^i = d^0 + d^1 + d^2 + \dots + d^h$$

$$\sum_{i=0}^h d^i = \frac{d^{h+1} - 1}{d - 1}, d > 1$$

sendo **h** a altura e **d** o número máximo de filhos de cada nó.

- a) Quantos nós tem a árvore?
- b) Qual é a raiz da árvore?
- c) Quais são os nós terminais?
- d) Qual o grau da árvore?
- e) Qual o nível da árvore?
- f) Quais são os nós descendentes do nó **D** ?
- g) Quais são os nós ancestrais do nó # ?
- h) Quais são nós irmãos?
- i) Há caminho entre os nós **C** e **W**?
- j) Qual o nível do nó **5**?
- k) Qual o grau do nó **A**?
- l) Quem é o pai do nó **K**?
- m) Qual o grau do nó **9**?



Tipo Abstrato de Dados (TAD) de Árvores

- O que o TAD Árvore deveria conter?
 - ✓ **Representação do tipo da árvore**
 - ✓ **Conjunto de operações que atuam sobre a árvore**
- Quais operações deveriam fazer parte desta TAD?

Tipo Abstrato de Dados (TAD) de Árvores

- O que o TAD Árvore deveria conter?
 - ✓ **Representação do tipo da árvore**
 - ✓ **Conjunto de operações que atuam sobre a árvore**
- Quais operações deveriam fazer parte desta TAD?

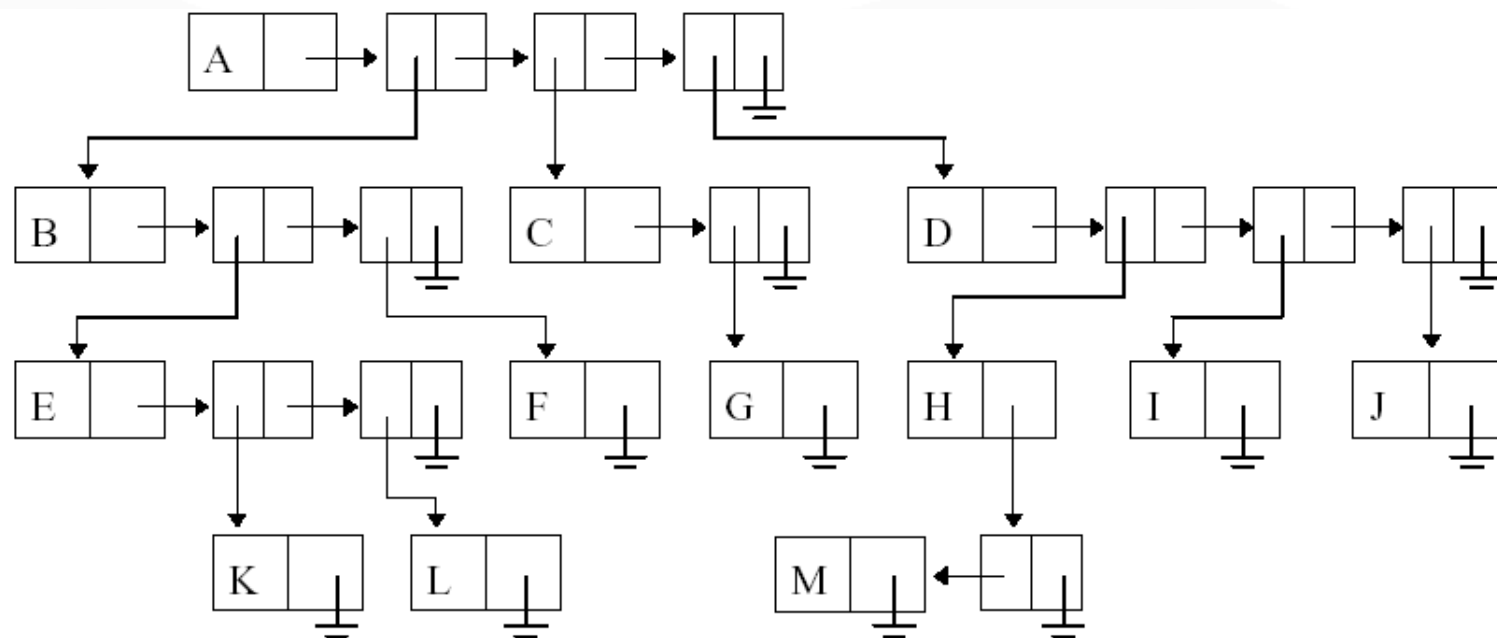
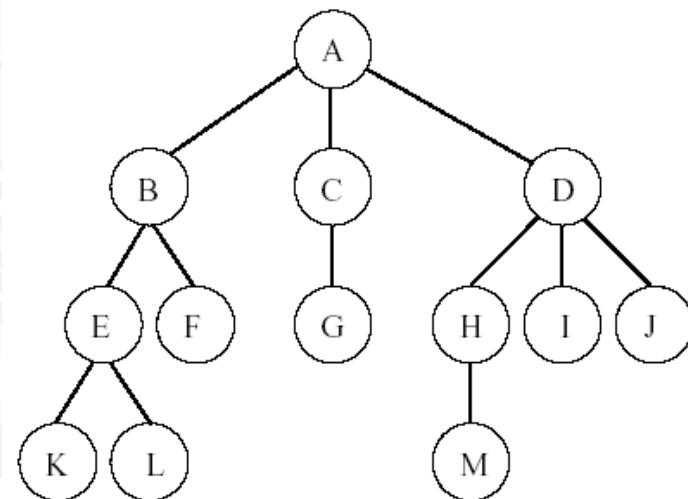
O conjunto de operações a ser definido depende de cada aplicação

TAD de Árvores (cont.)

➤ Estrutura utilizando apontadores

Dados	Apont. 1	Apont. 2	...	Apont. n
-------	----------	----------	-----	----------

✓ Exemplo:



TAD de Árvores (cont.)

- Um conjunto de operações necessário à uma maioria de aplicações é:
 1. Criar uma árvore sem descendentes
 2. Inserir um novo descendente em uma árvore
 3. Verificar se uma dada árvore possui descendentes
 4. Obter a lista de descendentes de uma dada árvore

TAD de Árvores (cont.)

```
typedef struct SNo *Tarvore;
```

```
typedef struct {  
    int Chave;  
    /* outros componentes */  
} TDados;
```

```
typedef struct SNo {  
    TDados Dados;  
    TLista Filhos;  
} Tno;
```