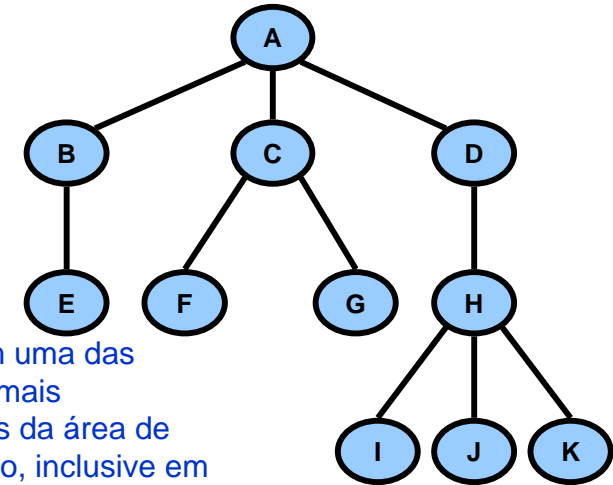


## Árvores

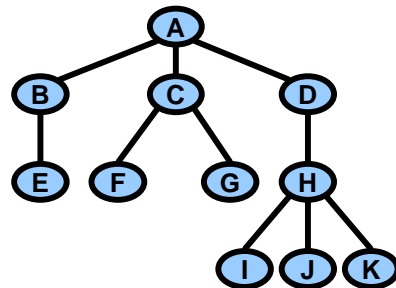
## Árvores



Constituem uma das estruturas mais importantes da área de computação, inclusive em aplicações

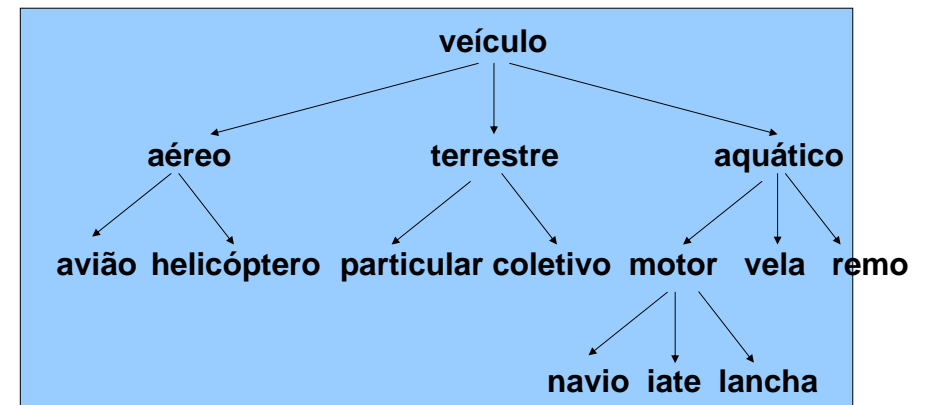
## Árvores

- Relacionamento Lógico
  - Hierarquia ou Subordinação
- Onde:
  - um subconjunto dos componentes é subordinado a outro



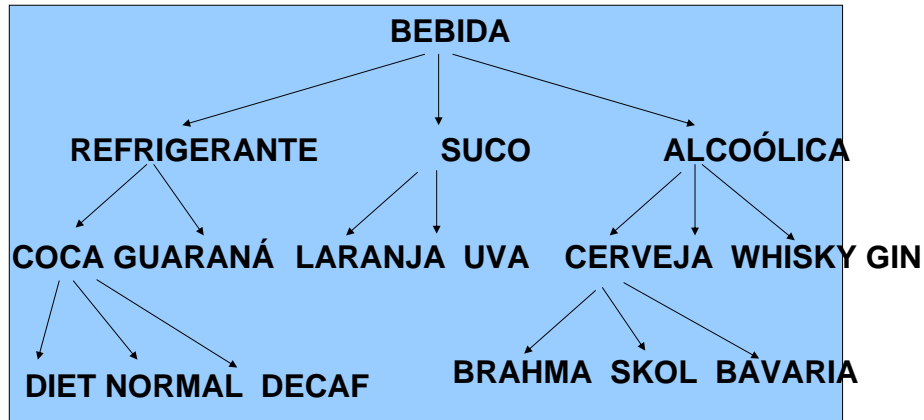
## Exemplos de Aplicações

- Hierarquia de **subordinação**
  - Classes e subclasses



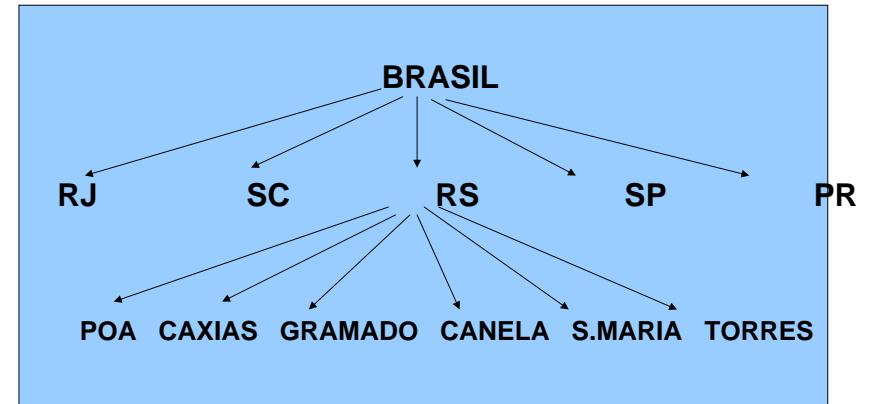
## Exemplos de Aplicações

- Hierarquia de **subordinação**
  - Classes e subclasses



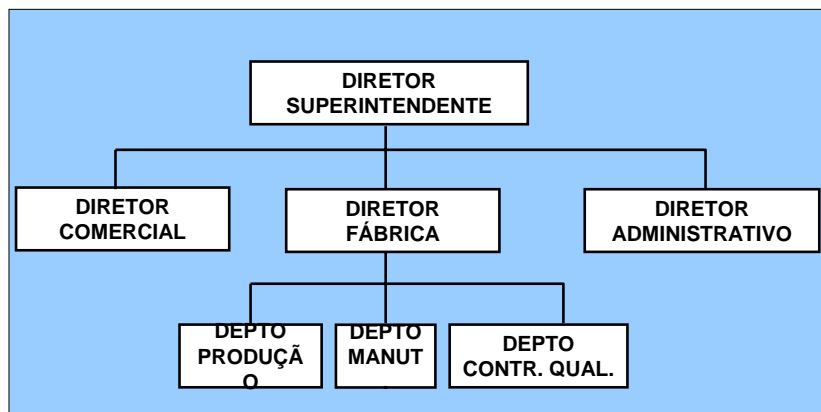
## Exemplos de Aplicações

- Abstração de composição



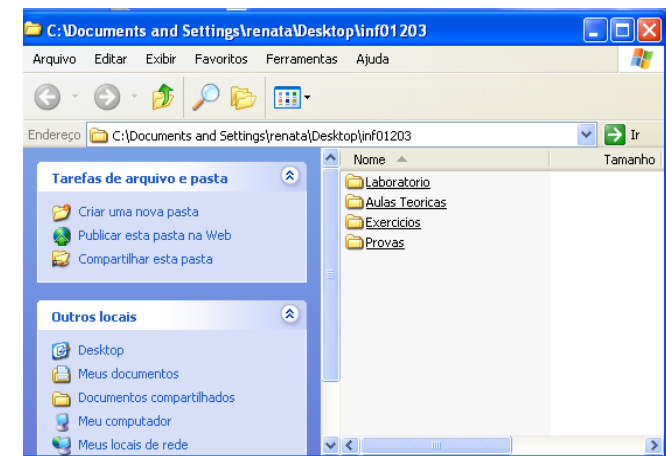
## Exemplos de Aplicações

- Organograma de empresa



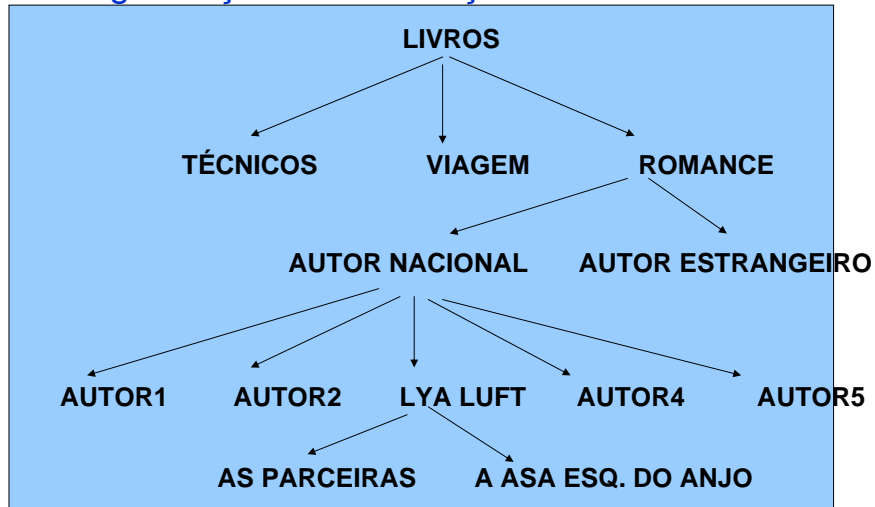
## Exemplos de Aplicações

- Diretórios e arquivos



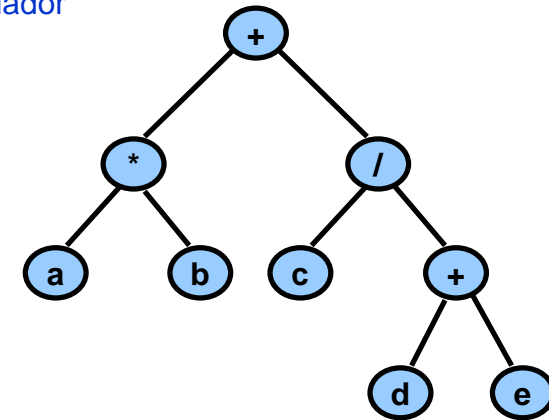
## Exemplos de Aplicações

- Organização de informações



## Exemplos de Aplicações

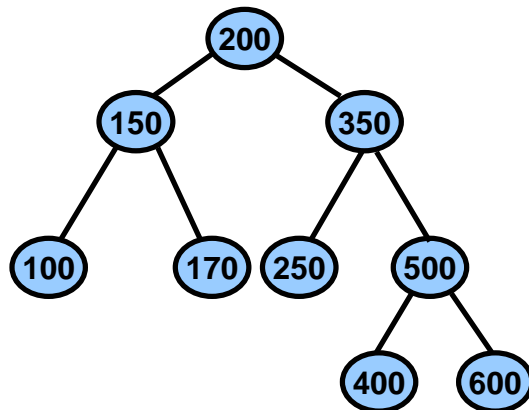
- Árvore de derivação  
– compilador



Expressão aritmética:  $(a * b) + (c / (d + e))$

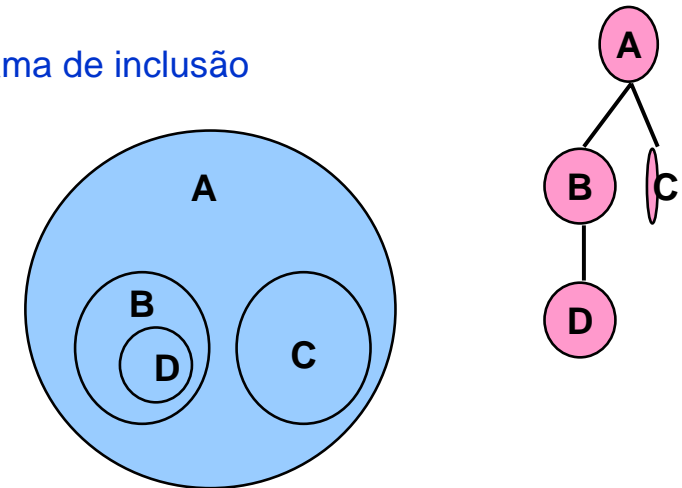
## Exemplos de Aplicações

- Ordenação de valores



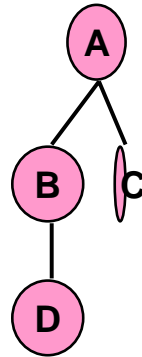
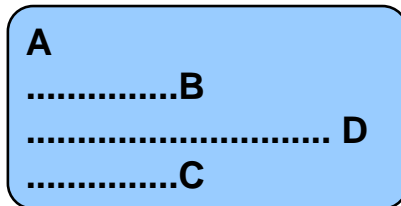
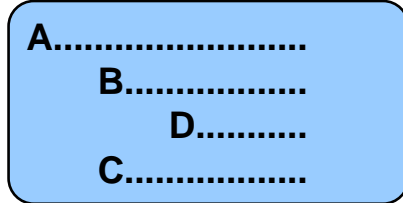
## Formas de Representação

- Diagrama de inclusão



## Formas de Representação

- Diagrama de barras



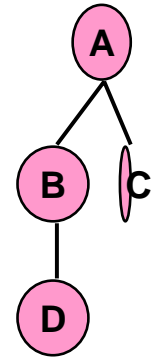
## Formas de Representação

- Níveis

1A; 1.1B; 1.1.1D; 1.2C

- Aninhamento

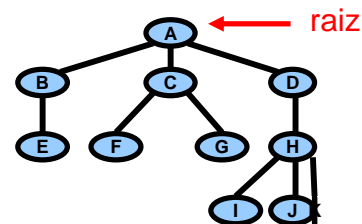
(A((B(D))(C)))



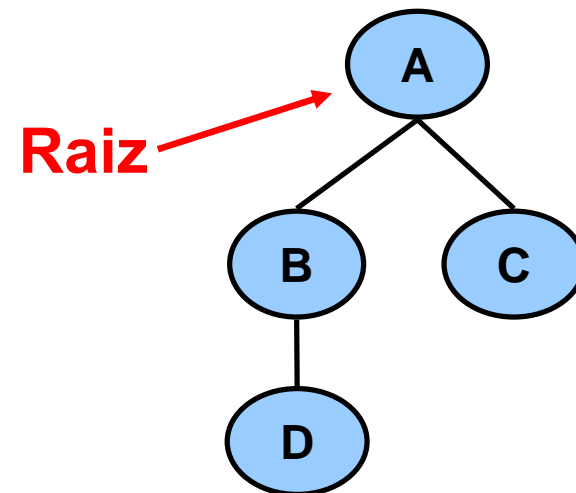
## Definição Formal

- Conjunto finito  $T$  de zero ou mais nós, tal que:

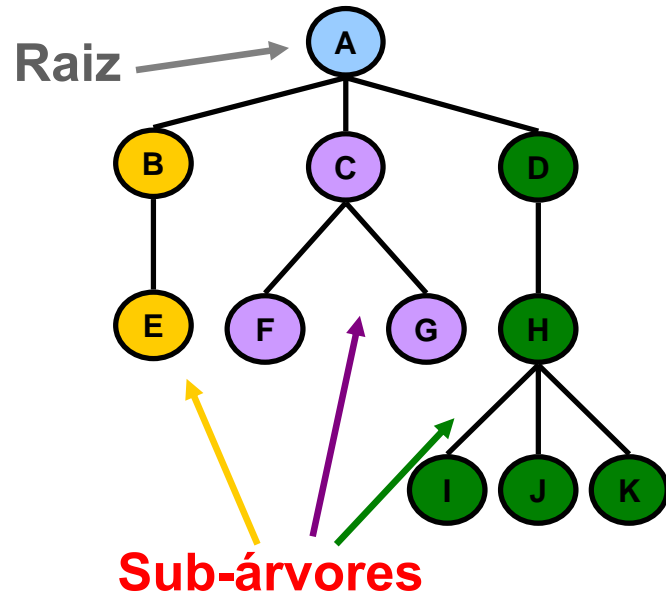
- número de nodos maior do que zero
  - existe um nó denominado **raiz** da árvore
  - os demais nós formam  $m > 0$  conjuntos disjuntos  $S_1, S_2, \dots, S_m$ , onde cada um destes é uma árvore ( $S_i$  são denominadas **sub-árvores**)
- número de nodos igual a zero
  - árvore vazia



## Terminologia



## Terminologia



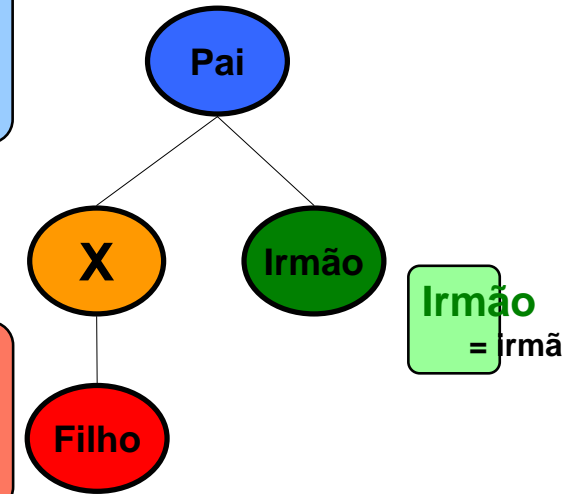
## Terminologia

### Pai

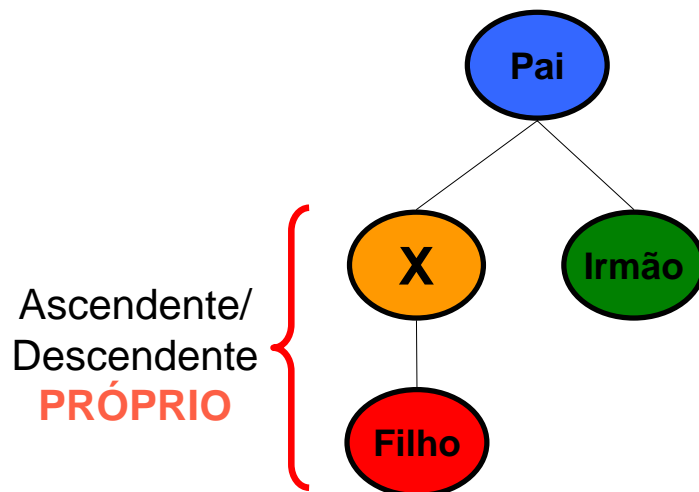
= mãe  
= ascendente  
= antecessor

### Filho

= filha  
= descendente  
= sucessor

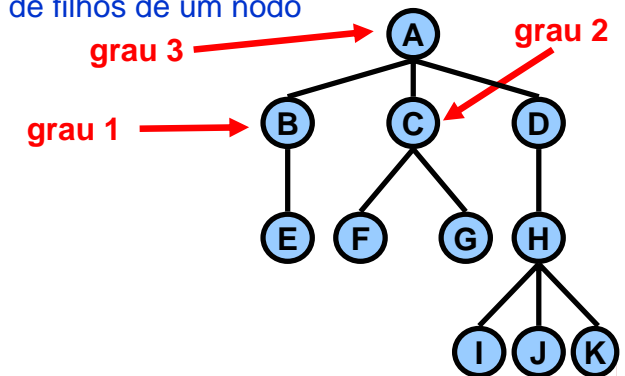


## Terminologia



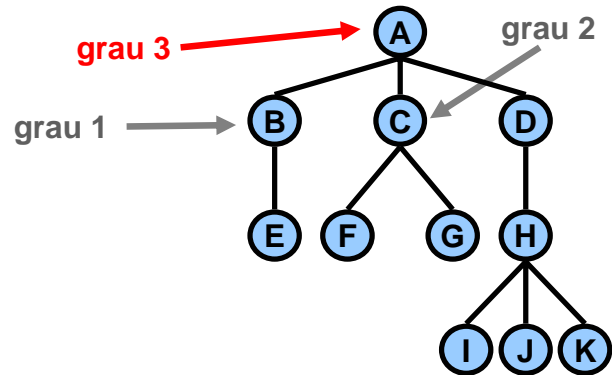
## Terminologia

- **Grau**
  - número de sub-árvores do nodo
- **Grau de saída**
  - número de filhos de um nodo



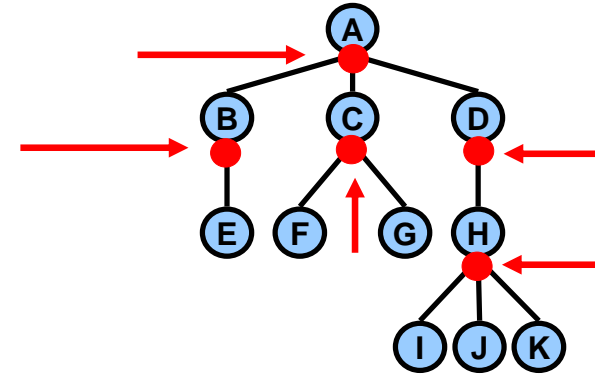
## Terminologia

- Grau de uma árvore
  - máximo entre os graus de seus nodos



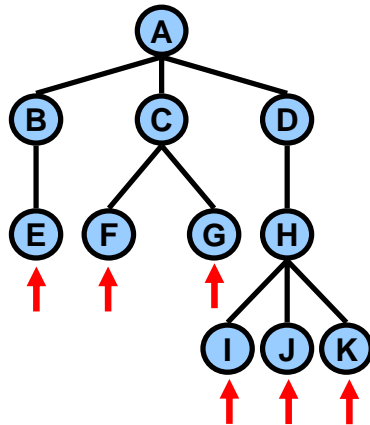
## Terminologia

- Nó de derivação (interno)
  - nó com grau maior do que zero



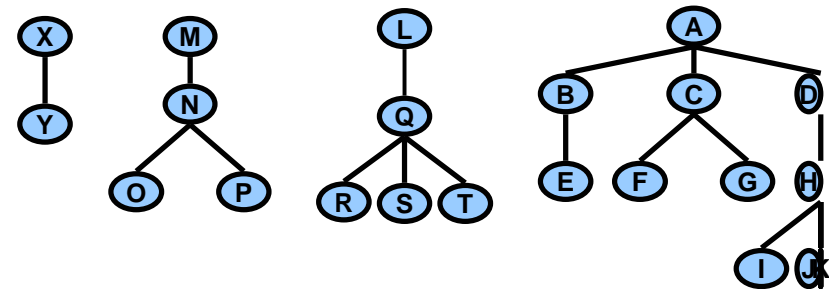
## Terminologia

- Nó folha (ou terminal ou externo)
  - grau igual a zero



## Terminologia

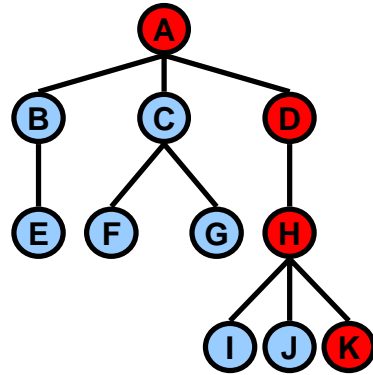
- Floresta



## Terminologia

- Caminho

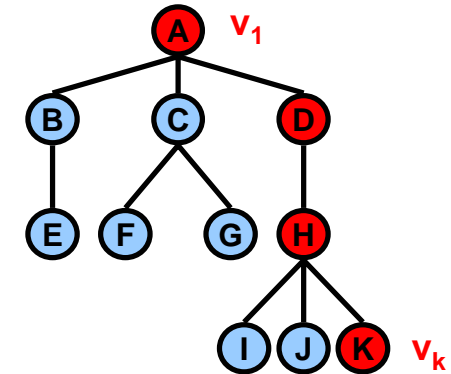
- sequência de nodos distintos, tal que existem sempre nodos consecutivos



## Terminologia

- Comprimento do caminho

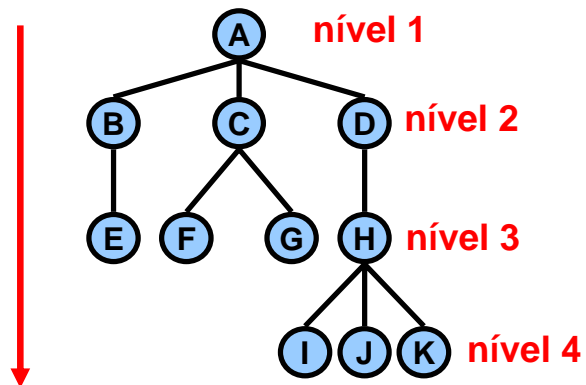
- sequência de nodos distintos, tal que
  - existem sempre nodos consecutivos
  - $v_1$  alcança  $v_k$  e  $v_k$  é alcançado por  $v_1$



## Terminologia

- Nível

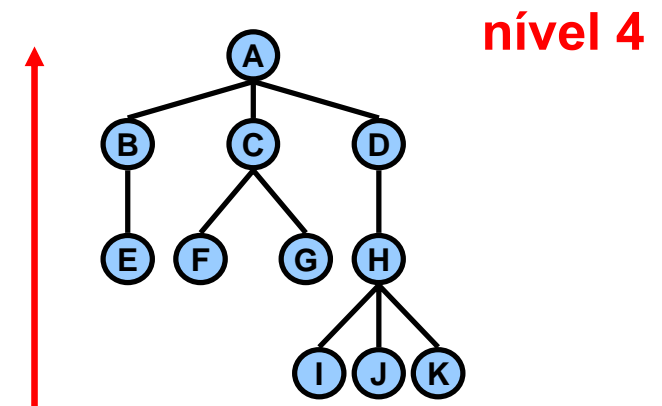
- número de ligações entre o nodo e a raiz, acrescido de uma unidade



## Terminologia

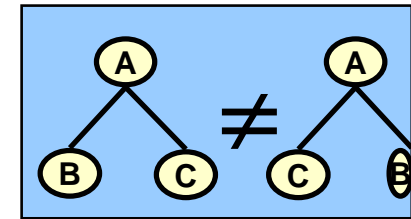
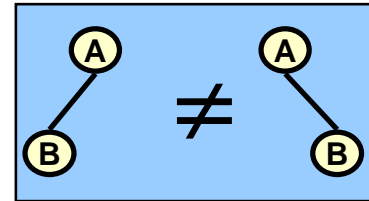
- Altura (profundidade)

- maior nível



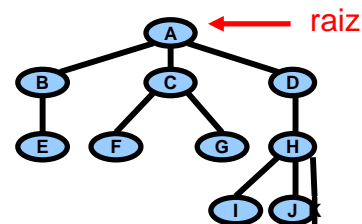
## Terminologia

- **Árvore ordenada**
  - ordem das sub-árvores é relevante



## Definição Formal

- Conjunto finito **T** de zero ou mais nós, tal que:
  - número de nodos maior do que zero
    - existe um nó denominado **raiz** da árvore
    - os demais nós formam  $m > 0$  conjuntos disjuntos  $S_1, S_2, \dots, S_m$ , onde cada um destes é uma árvore ( $S_i$  são denominadas **sub-árvores**)
  - número de nodos igual a zero
    - **árvore vazia**



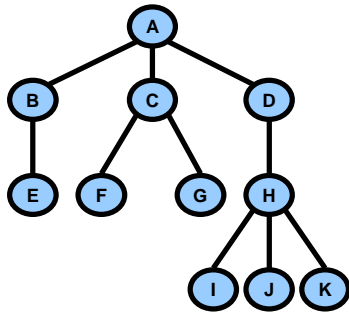
## Operações sobre as Árvores

- **Dados**
  - árvore **A**
- **Operações**
  - criação da árvore
  - inserção de um novo nodo
    - raiz
    - folha
    - posição intermediária
  - exclusão de um determinado nodo
  - acesso a um nodo
    - determinar forma de percorrer a árvore
  - destruição da árvore



# Árvores

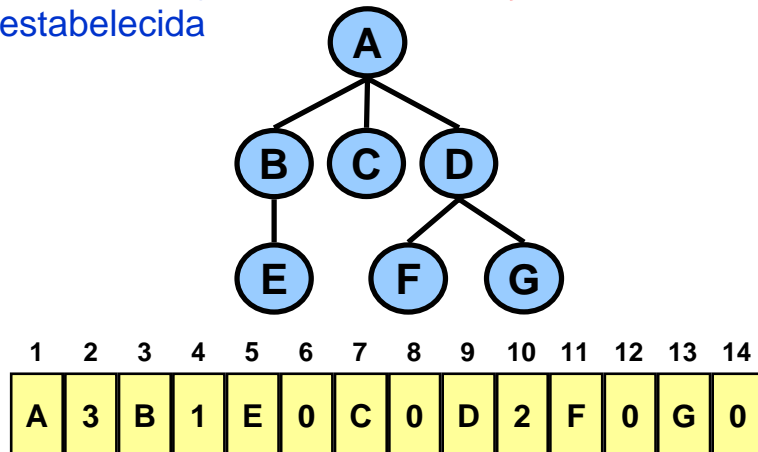
- Como armazenar os nós de uma árvore?
- Como percorrer uma árvore?
- Como inserir novos nós?
- Como excluir nós?



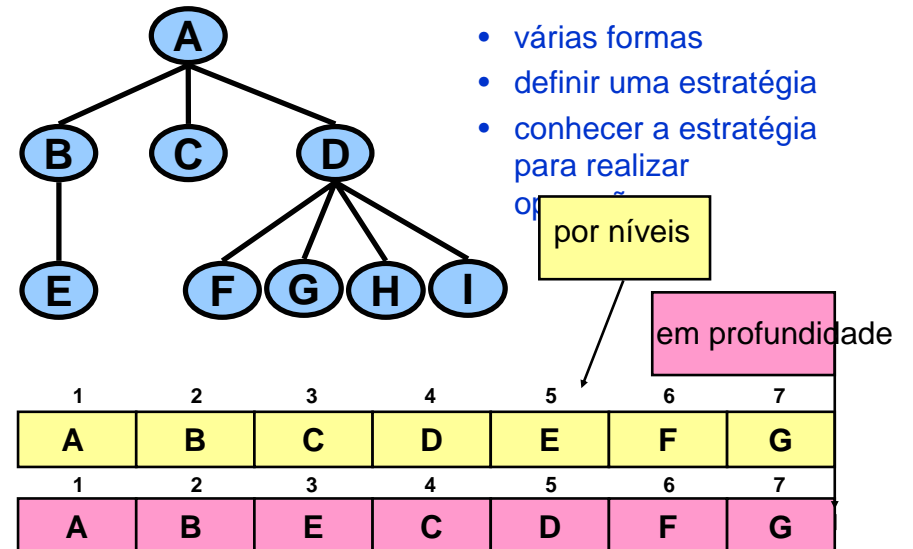
## Árvores Implementadas por Contigüidade Física

## Árvores – Contigüidade Física

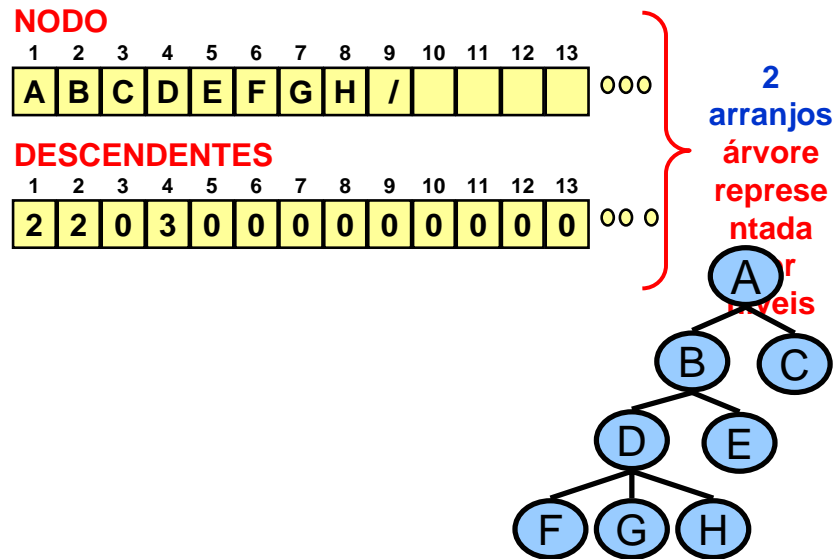
- Os nodos da árvore são dispostos na memória segundo uma **convenção** estabelecida



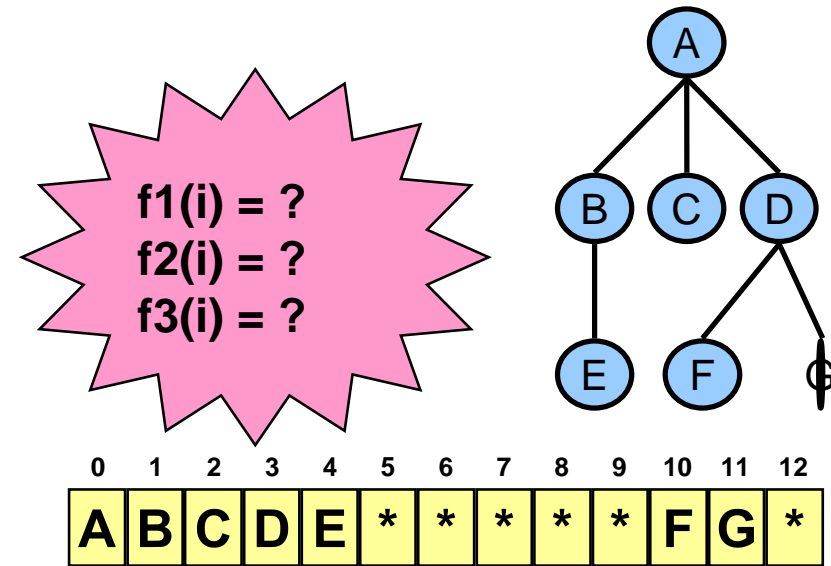
## Árvores – Contigüidade Física



## Árvores – Contigüidade Física



## Árvore Ternária



## TAD Árvore – contigüidade física



## TAD Árvore – contigüidade física

- Árvore representada por níveis
  - alInfo: Array [1..n] de Info;
  - aFilhos: Array [1..n] de inteiro;
  - Descritor da árvore (descA)
    - raiz: inteiro; // endereço da raiz no array
    - numNodos: inteiro;
    - altura: inteiro;
    - grauMax: inteiro;

## TAD - operações

```
procedure inicializa(var a: descA);  
  // inicializa a árvore  
procedure insere(var a: descA; var al: aInfo; var  
  a2: aFilhos; dado: Info; pai:  
  Info);  
  // insere um nodo na árvore  
procedure remove(var a: descA; var al: aInfo; var  
  a2: aFilhos; dado: Info);  
  // remove um nodo da árvore  
function localiza(a: descA; al: aInfo; a2:  
  aFilhos; dado: Info):inteiro;  
  // busca o campo de informação de um nodo e  
  // retorna o índice do array  
procedure visita(a: descA; al: aInfo; a2:  
  aFilhos);  
  // percorre todos os nodos da árvore  
procedure destroi(var a: descA);  
  // destrói a árvore
```

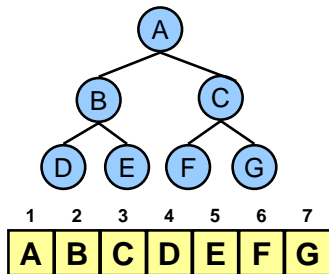
## Análise – Contigüidade Física

- não constitui, em geral, uma boa solução
- dificuldades para manipulação da estrutura (hierarquia)
- geralmente eficiente em termos de espaço ocupado

## Análise – Contigüidade Física

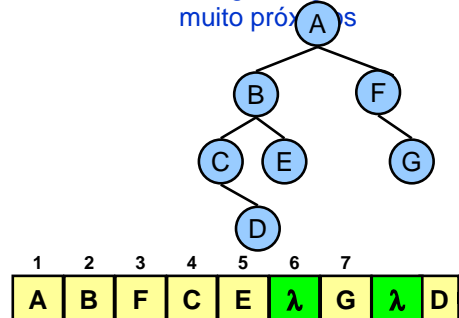
### • Exceção 01

- quando os nodos são processados exatamente na mesma ordem em que são armazenados



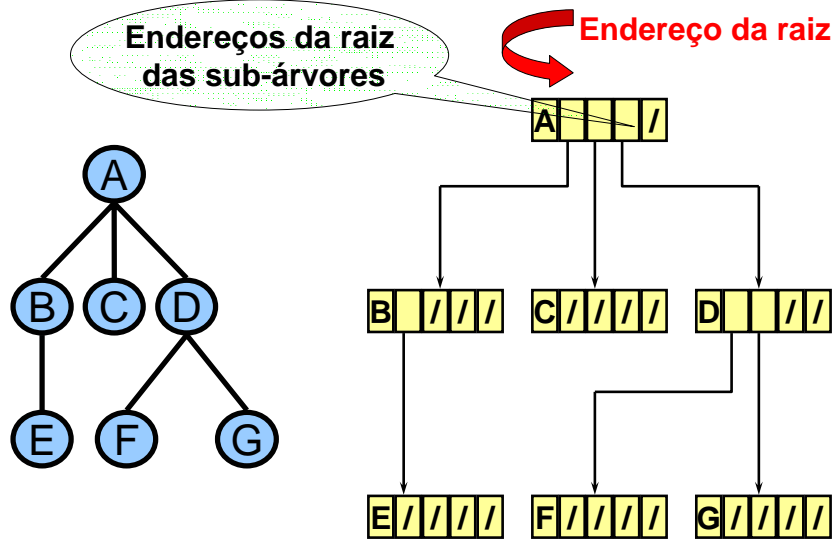
### • Exceção 02

- quando, excetuando-se as folhas, que aparecem no mesmo nível, os demais nodos têm graus iguais ou muito próximos



## Árvores Implementadas por Encadeamento

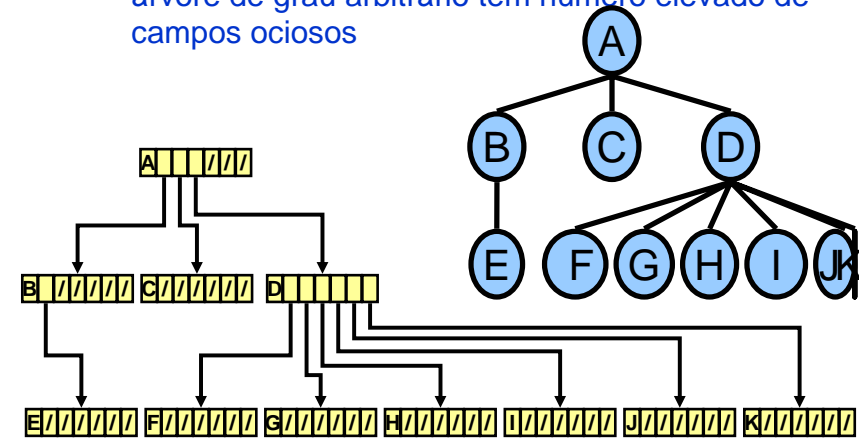
## Árvores - Encadeamento



## Árvores - Encadeamento

- Problema:

- árvore de grau arbitrário tem número elevado de campos ociosos



## TAD – Árvores por Encadeamento



## TAD Árvore – contigüidade física

- nodoA // Nodo da árvore
  - dado: Info
  - filho1: ^nodoA
  - filho2: ^nodoA
  - ...
  - filhoN: ^nodoA
- Descritor da árvore (descA)
  - raiz: ^nodoA; // endereço da raiz no array
  - numNodos: inteiro;
  - altura: inteiro;
  - grauMax: inteiro;

## Análise – Encadeamento

- problemas quando grau dos nodos é variado – muitos elos vazios
- acesso somente através de raiz – disciplina para percorrer árvore
- inserção e remoção são simplificadas
- hierarquia é intuitiva – esta forma de implementação é adotada a partir de agora