

# Árvores - Conceitos Básicos

Prof. Luiz Gustavo Almeida Martins

# Árvores: conceitos básicos

---

**Árvore** é uma **estrutura não linear** usada para representar relações de **hierarquia ou subordinação**

Pode ser definida como um **grafo acíclico e conexo**

Pares de vértice podem ser ligados por uma **única aresta**

# Árvores: conceitos básicos

---

**Árvore** é uma **estrutura não linear** usada para representar relações de **hierarquia ou subordinação**

Pode ser definida como um **grafo acíclico e conexo**

Pares de vértice podem ser ligados por uma **única aresta**

Formado por **vértices (nós) e arestas (arcos ou ramos)**:

# Árvores: conceitos básicos

---

**Árvore** é uma **estrutura não linear** usada para representar relações de **hierarquia ou subordinação**

Pode ser definida como um **grafo acíclico e conexo**

Pares de vértice podem ser ligados por uma **única aresta**

Formado por **vértices (nós) e arestas (arcos ou ramos)**:

**Vértice:** elemento que pode armazenar qualquer tipo de informação

Representado por um **círculo**

# Árvores: conceitos básicos

---

**Árvore** é uma **estrutura não linear** usada para representar relações de **hierarquia ou subordinação**

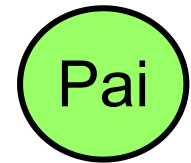
Pode ser definida como um **grafo acíclico e conexo**

Pares de vértice podem ser ligados por uma **única aresta**

Formado por **vértices (nós) e arestas (arcos ou ramos)**:

**Vértice:** elemento que pode armazenar qualquer tipo de informação

Representado por um **círculo**



# Árvores: conceitos básicos

**Árvore** é uma **estrutura não linear** usada para representar relações de **hierarquia ou subordinação**

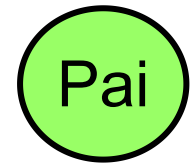
Pode ser definida como um **grafo acíclico e conexo**

Pares de vértice podem ser ligados por uma **única aresta**

Formado por **vértices (nós) e arestas (arcos ou ramos)**:

**Vértice:** elemento que pode armazenar qualquer tipo de informação

Representado por um **círculo**



**Aresta:** ligação entre 2 vértices que define uma **relação de parentesco (hierarquia)**

Representada por uma **linha**

# Árvores: conceitos básicos

**Árvore** é uma **estrutura não linear** usada para representar relações de **hierarquia ou subordinação**

Pode ser definida como um **grafo acíclico e conexo**

Pares de vértice podem ser ligados por uma **única aresta**

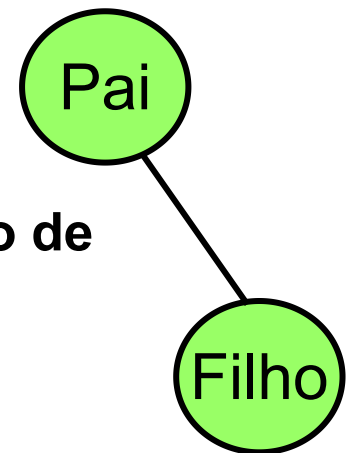
Formado por **vértices (nós) e arestas (arcos ou ramos)**:

**Vértice:** elemento que pode armazenar qualquer tipo de informação

Representado por um **círculo**

**Aresta:** ligação entre 2 vértices que define uma **relação de parentesco (hierarquia)**

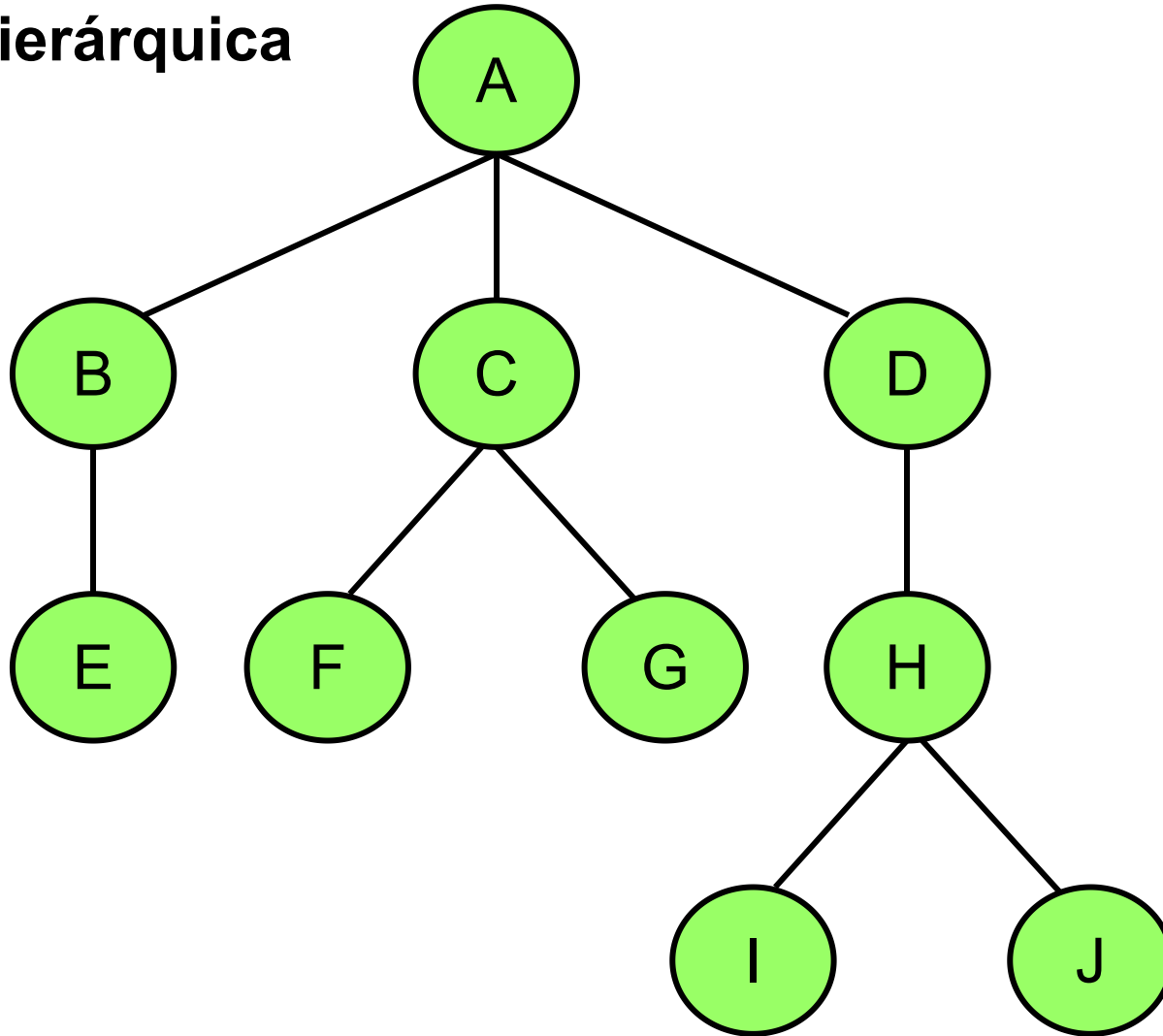
Representada por uma **linha**



# Árvores: representação gráfica

---

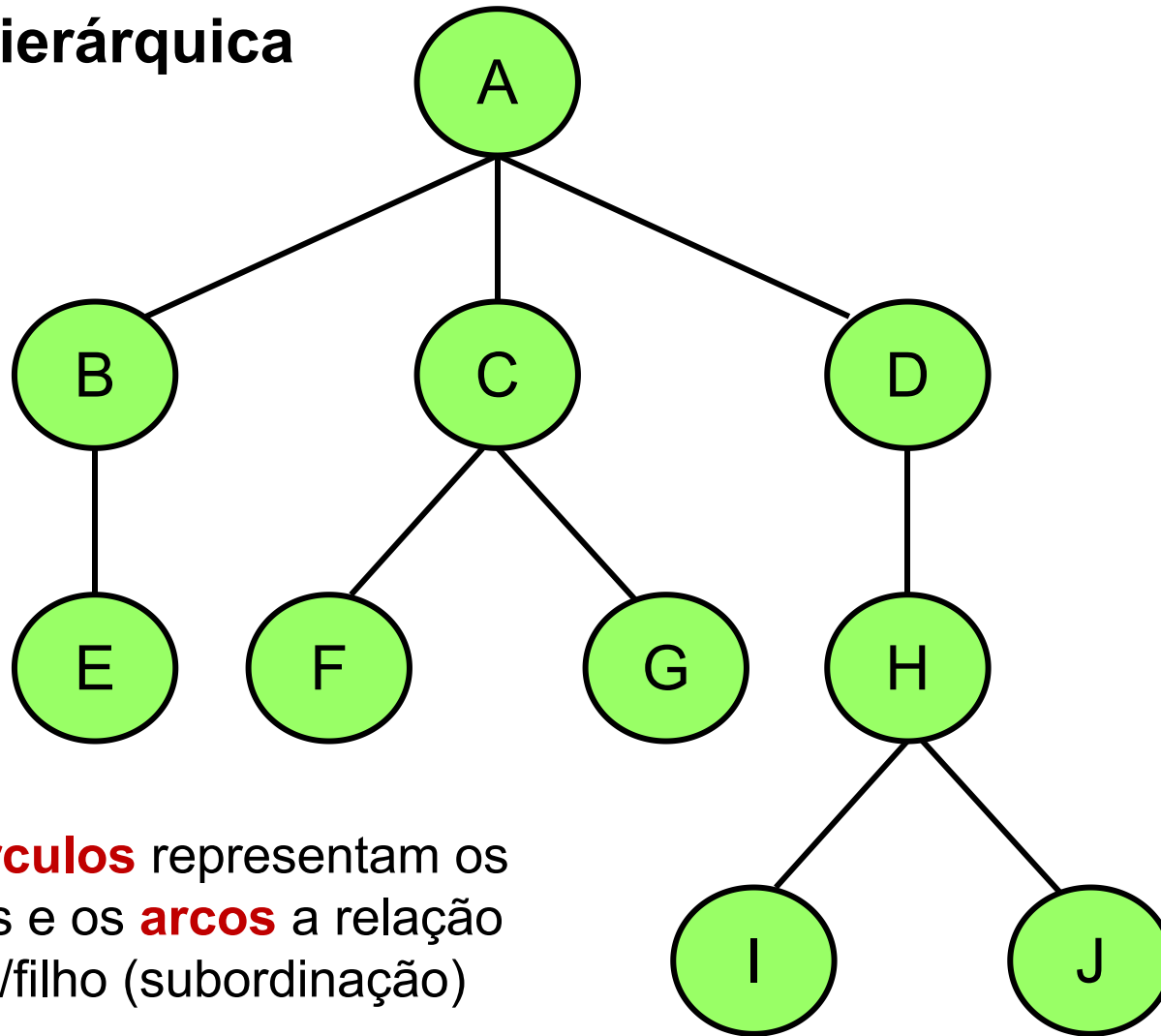
## Hierárquica





# Árvores: representação gráfica

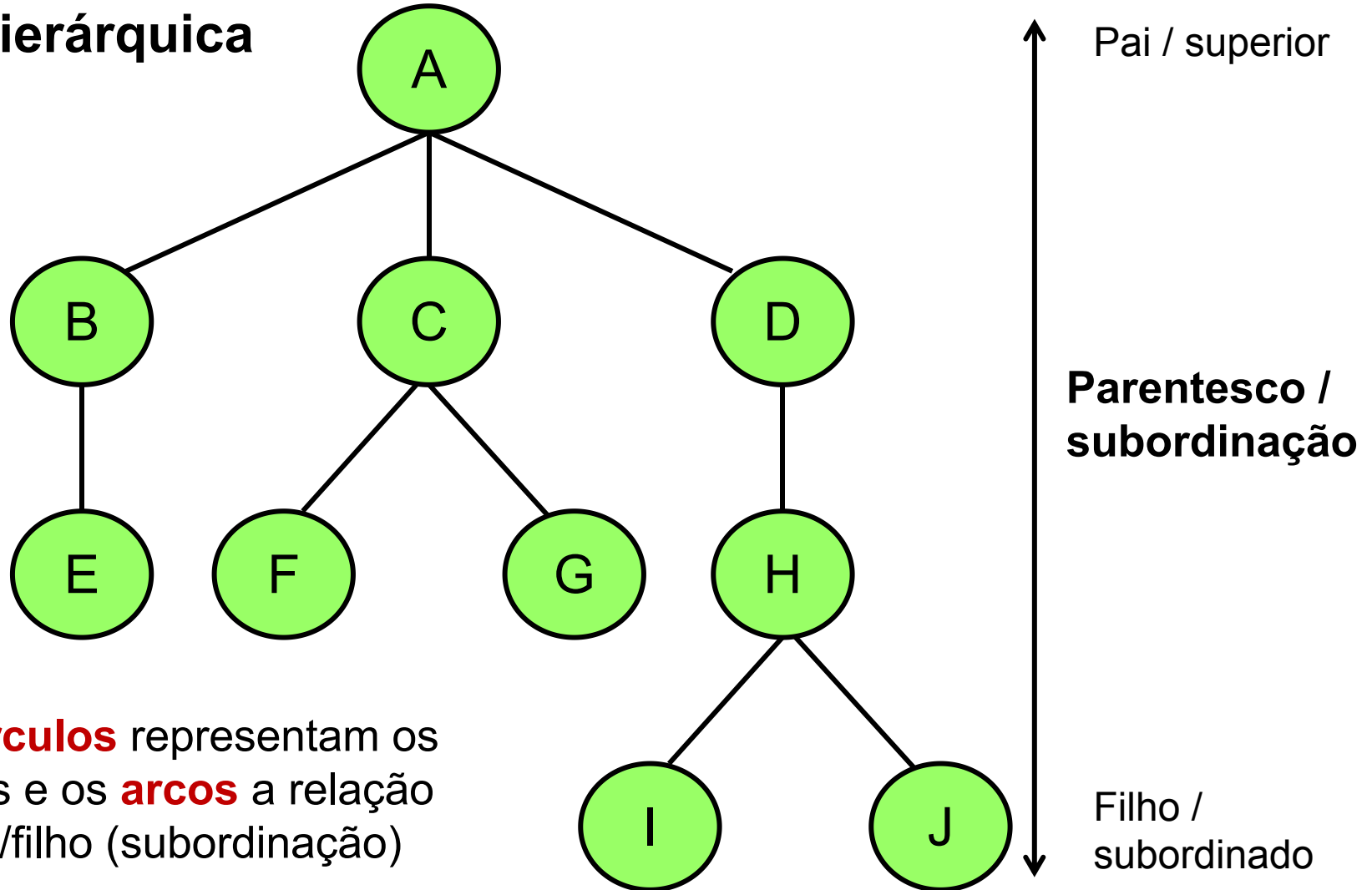
## Hierárquica



**Círculos** representam os nós e os **arcos** a relação pai/filho (subordinação)

# Árvores: representação gráfica

**Hierárquica**

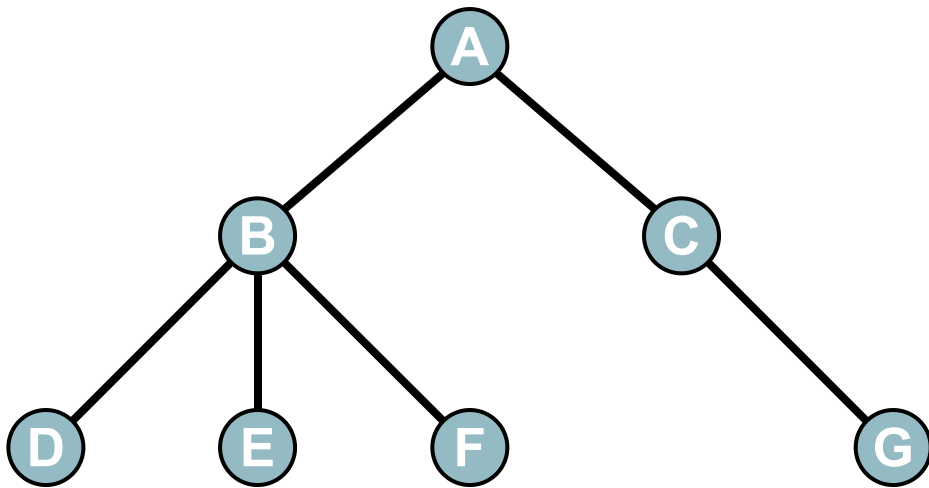


**Círculos** representam os nós e os **arcos** a relação pai/filho (subordinação)

# Árvores: outras formas de representação

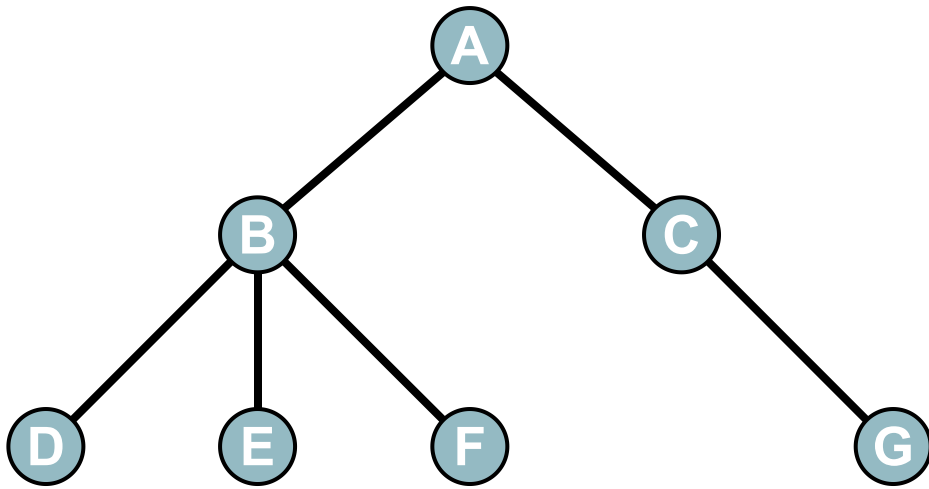
---

## Hierárquica

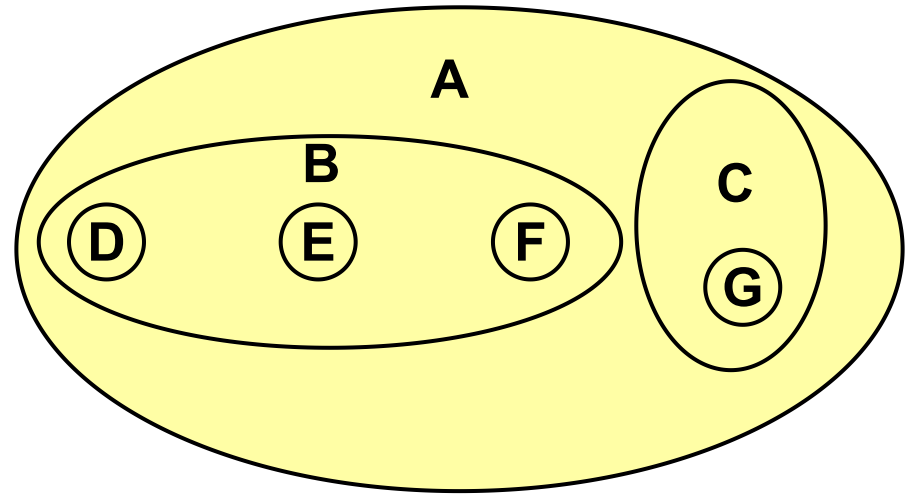


# Árvores: outras formas de representação

**Hierárquica**

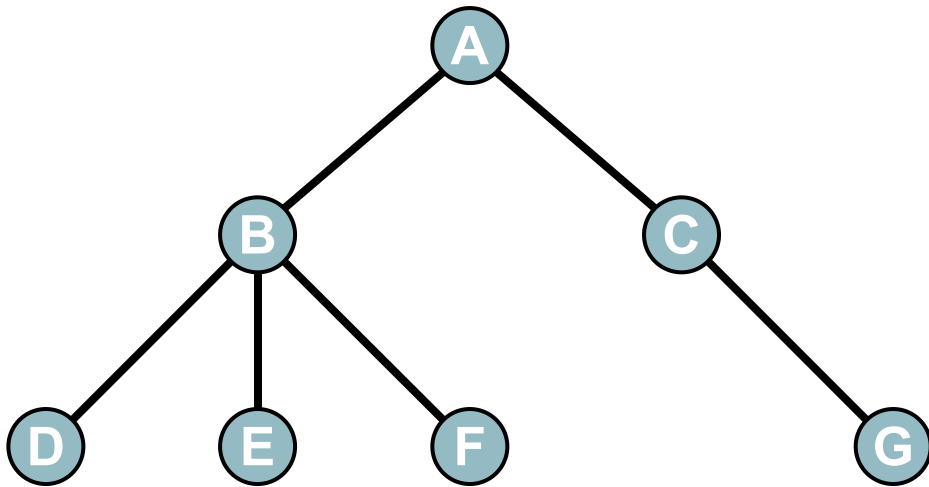


**Diagrama de inclusão (Venn)**

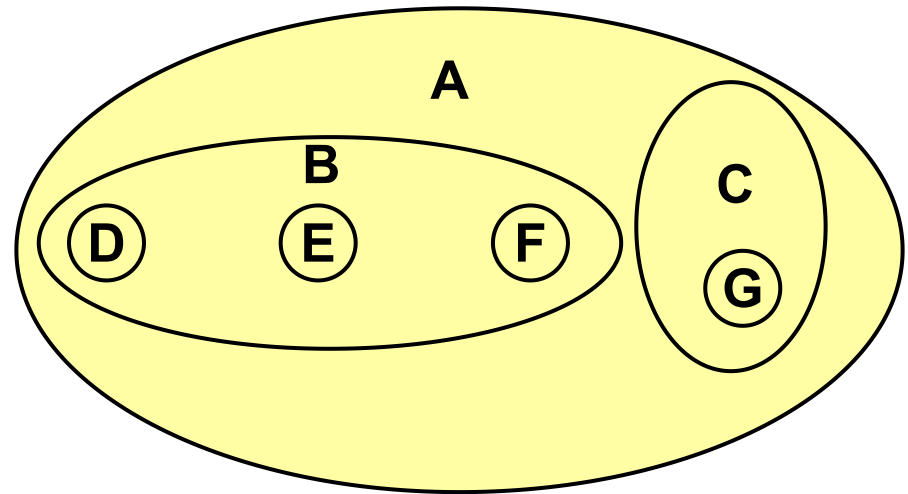


# Árvores: outras formas de representação

## Hierárquica



## Diagrama de inclusão (Venn)



## Representação aninhada

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

# Árvores: conceitos básicos

---

Estrutura utilizada em diversas áreas

**Ex:** Biologia, administração, medicina, etc.

# Árvores: conceitos básicos

---

Estrutura utilizada em diversas áreas

**Ex:** Biologia, administração, medicina, etc.

Possui grande importância para a computação

# Árvores: conceitos básicos

---

Estrutura utilizada em diversas áreas

**Ex:** Biologia, administração, medicina, etc.

Possui grande importância para a **computação**

Seu uso pode melhorar significativamente a eficiência dos algoritmos



# Árvores: conceitos básicos

---

Estrutura utilizada em diversas áreas

**Ex:** Biologia, administração, medicina, etc.

Possui grande importância para a computação

Seu uso pode melhorar significativamente a eficiência dos algoritmos

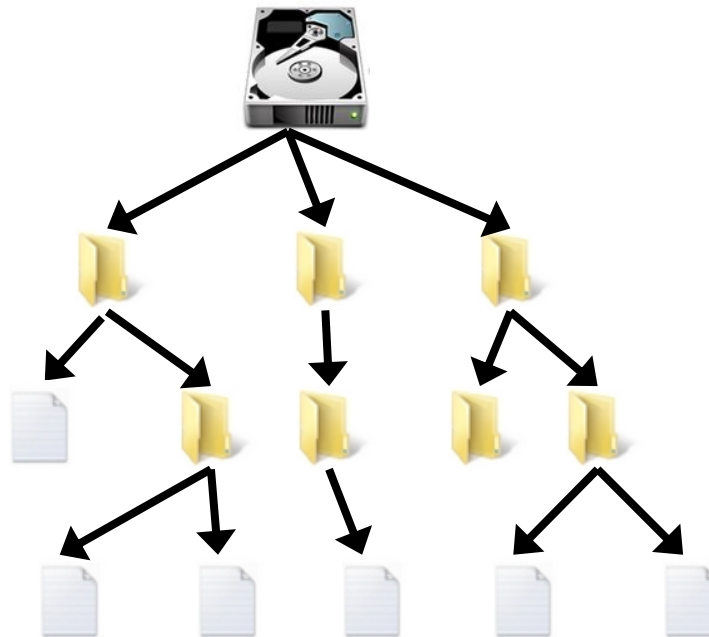
Empregada em muitos tipos de aplicações

**Ex:** Banco de dados, compiladores, compactadores, inteligência artificial, etc.

# Árvores: exemplos de aplicação

---

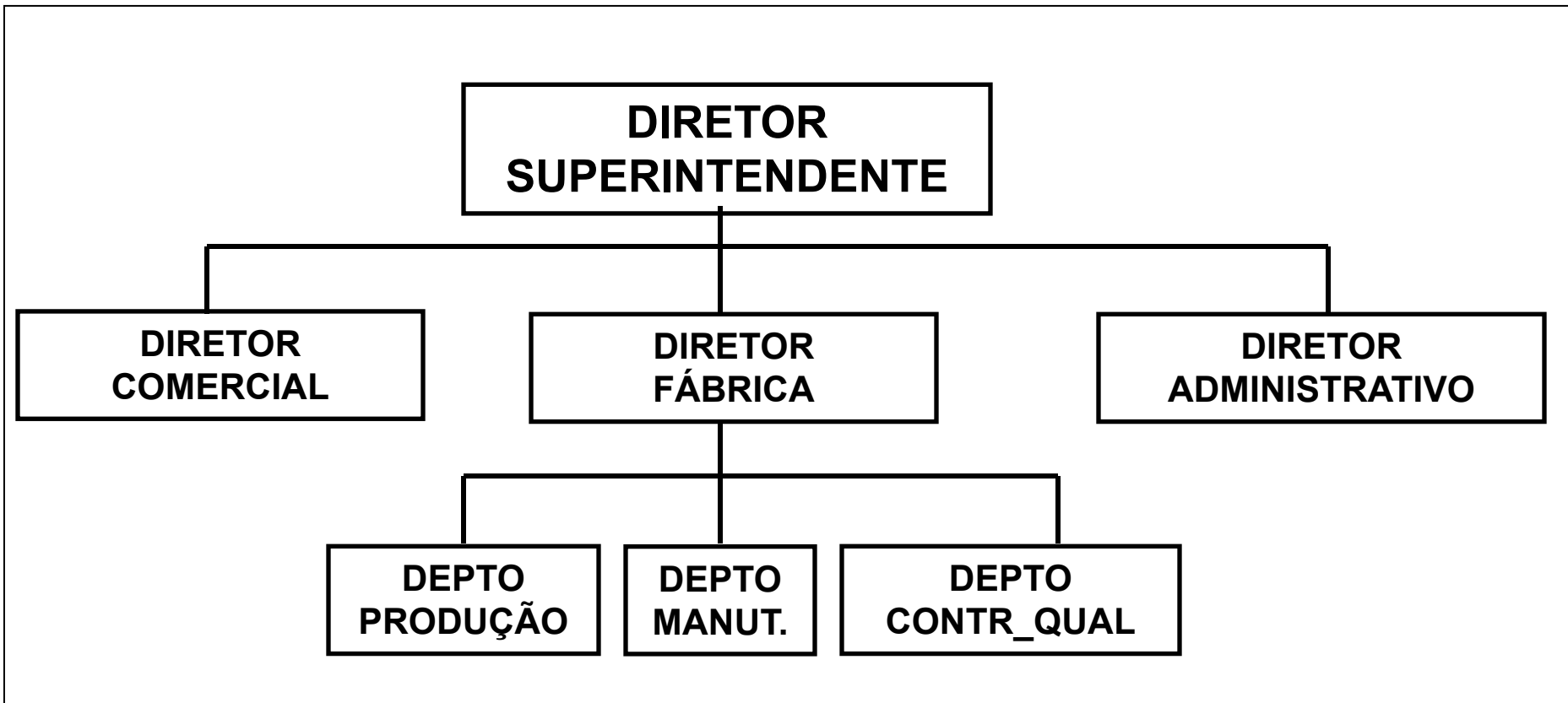
## **Hierarquia de dependência (organização da estrutura de diretórios e arquivos)**



# Árvores: exemplos de aplicação

---

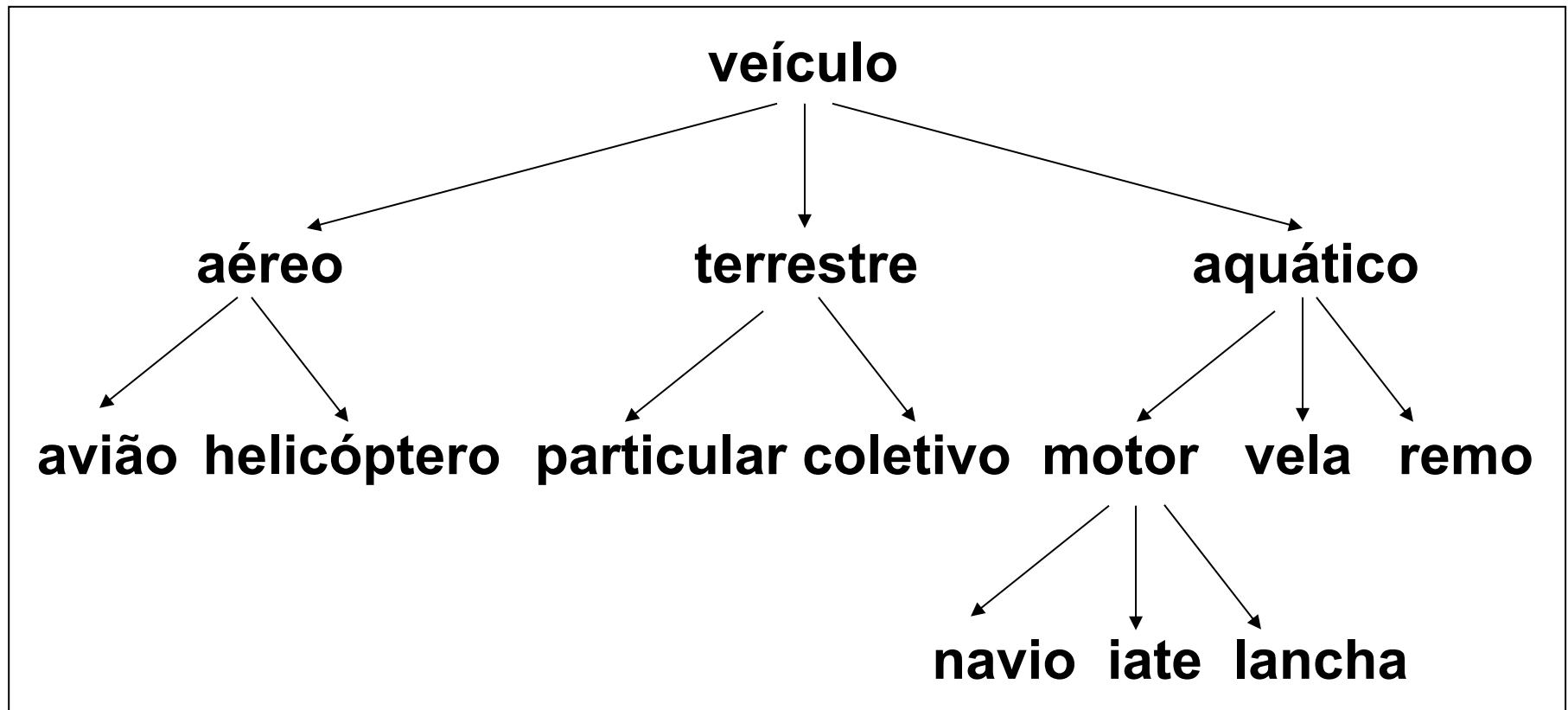
## Hierarquia de dependência (organograma de empresa)



# Árvores: exemplos de aplicação

---

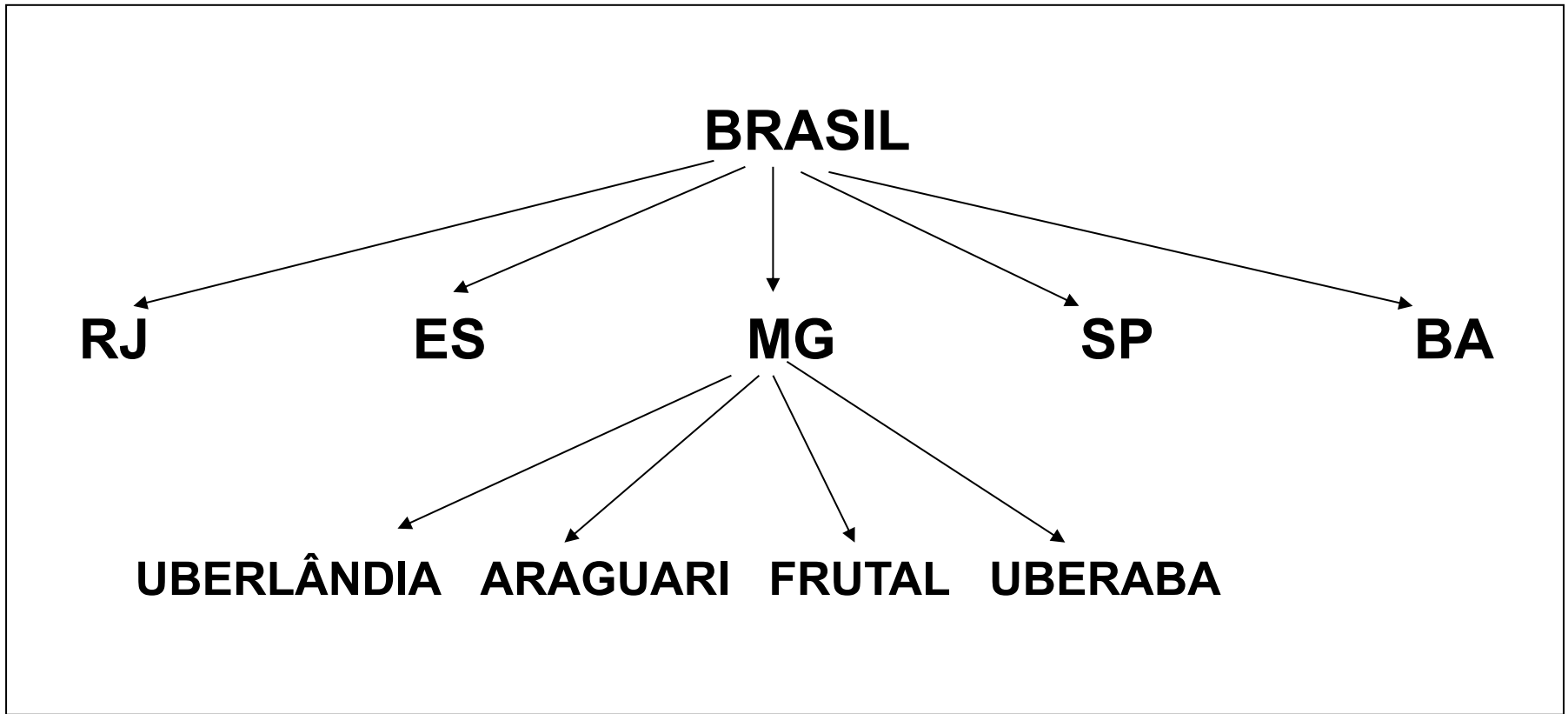
## Hierarquia de especialização



# Árvores: exemplos de aplicação

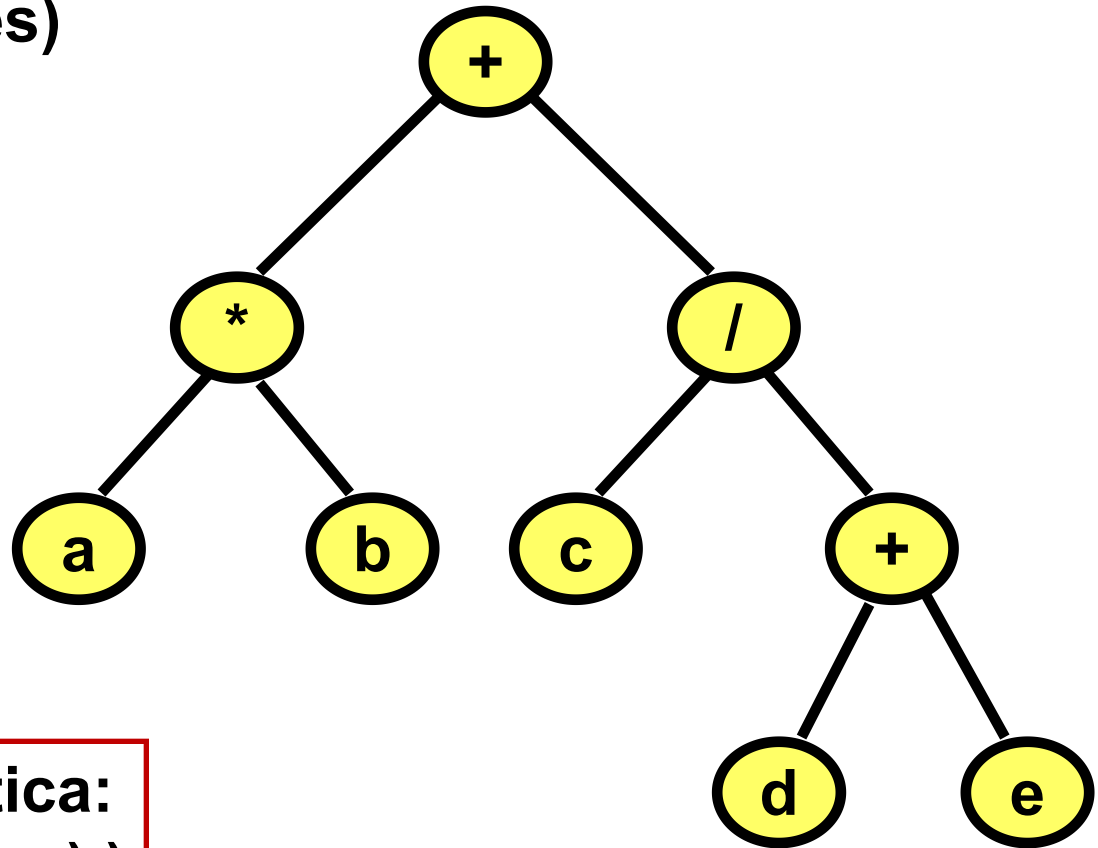
---

## Hierarquia de composição



# Árvores: exemplos de aplicação

## Árvore de derivação (compiladores)

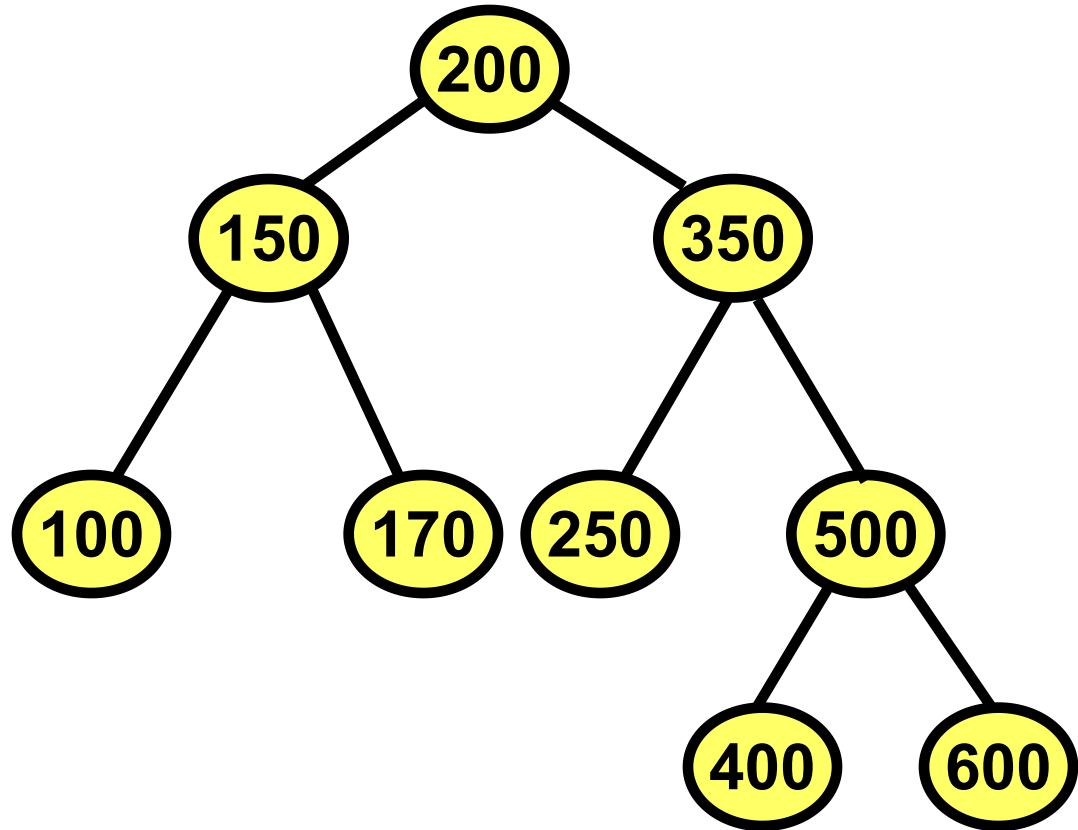


**Expressão aritmética:**  
 **$(a * b) + (c / (d + e))$**

# Árvores: exemplos de aplicação

---

## Ordenação e busca



# Árvores: terminologias

---

**Relação de parentesco entre nós:**



# Árvores: terminologias

---

## Relação de parentesco entre nós:

**Pai/filho:**  $A$  é **pai** de  $B$ , se existe um arco entre eles e  $A$  está acima de  $B$ . Neste caso,  $B$  é **filho** de  $A$

# Árvores: terminologias

---

## Relação de parentesco entre nós:

**Pai/filho:**  $A$  é **pai** de  $B$ , se existe um arco entre eles e  $A$  está acima de  $B$ . Neste caso,  $B$  é **filho** de  $A$

**Pai** é vértice **imediatamente superior** na hierarquia

# Árvores: terminologias

---

## Relação de parentesco entre nós:

**Pai/filho:**  $A$  é **pai** de  $B$ , se existe um arco entre eles e  $A$  está acima de  $B$ . Neste caso,  $B$  é **filho** de  $A$

**Pai** é vértice **imediatamente superior** na hierarquia

**Filhos** são os vértices **imediatamente abaixo** de um nó específico

# Árvores: terminologias

---

## Relação de parentesco entre nós:

**Pai/filho:**  $A$  é **pai** de  $B$ , se existe um arco entre eles e  $A$  está acima de  $B$ . Neste caso,  $B$  é **filho** de  $A$

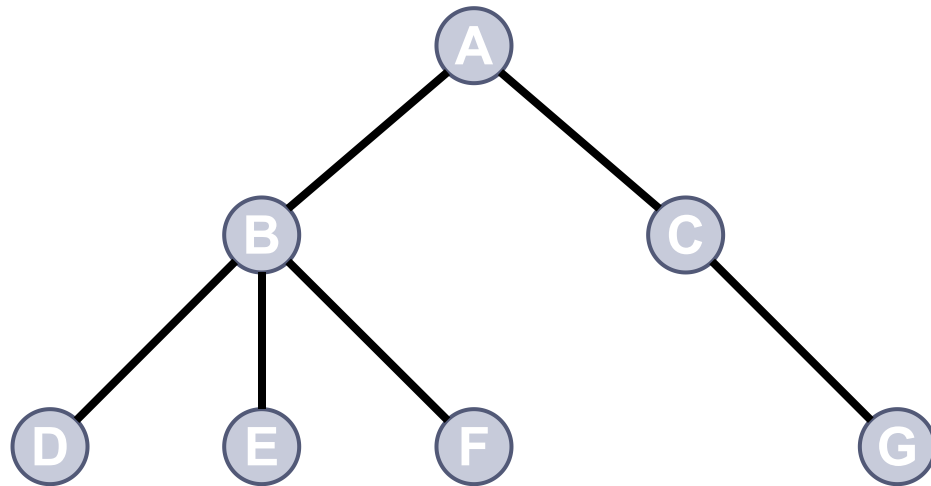
**Pai** é vértice **imediatamente superior** na hierarquia

**Filhos** são os vértices **imediatamente abaixo** de um nó específico

**Irmão:**  $A$  é **irmão** de  $B$ , se  $A$  e  $B$  são filhos do mesmo pai

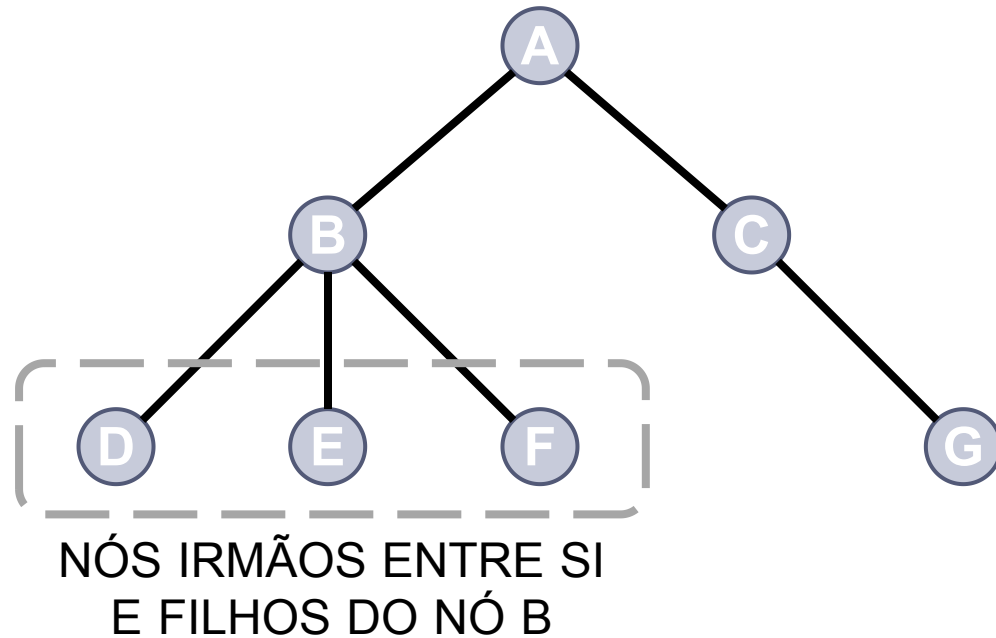
# Árvores: exemplo

---



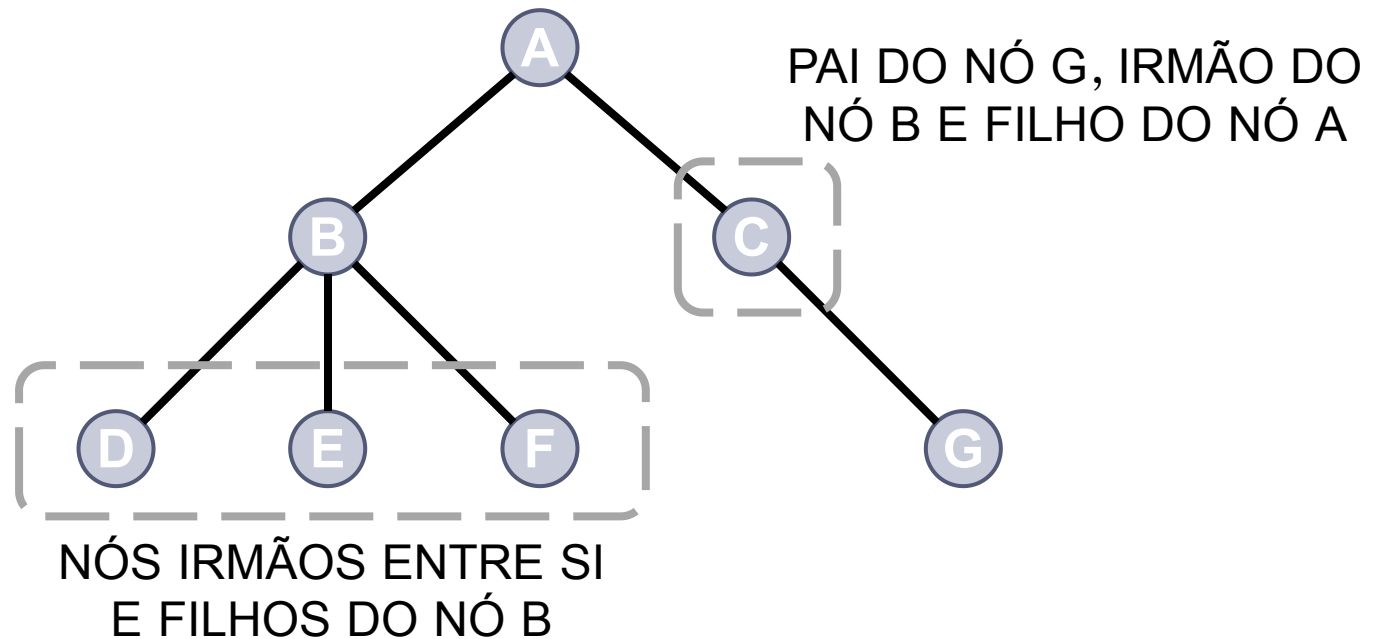
# Árvores: exemplo

---



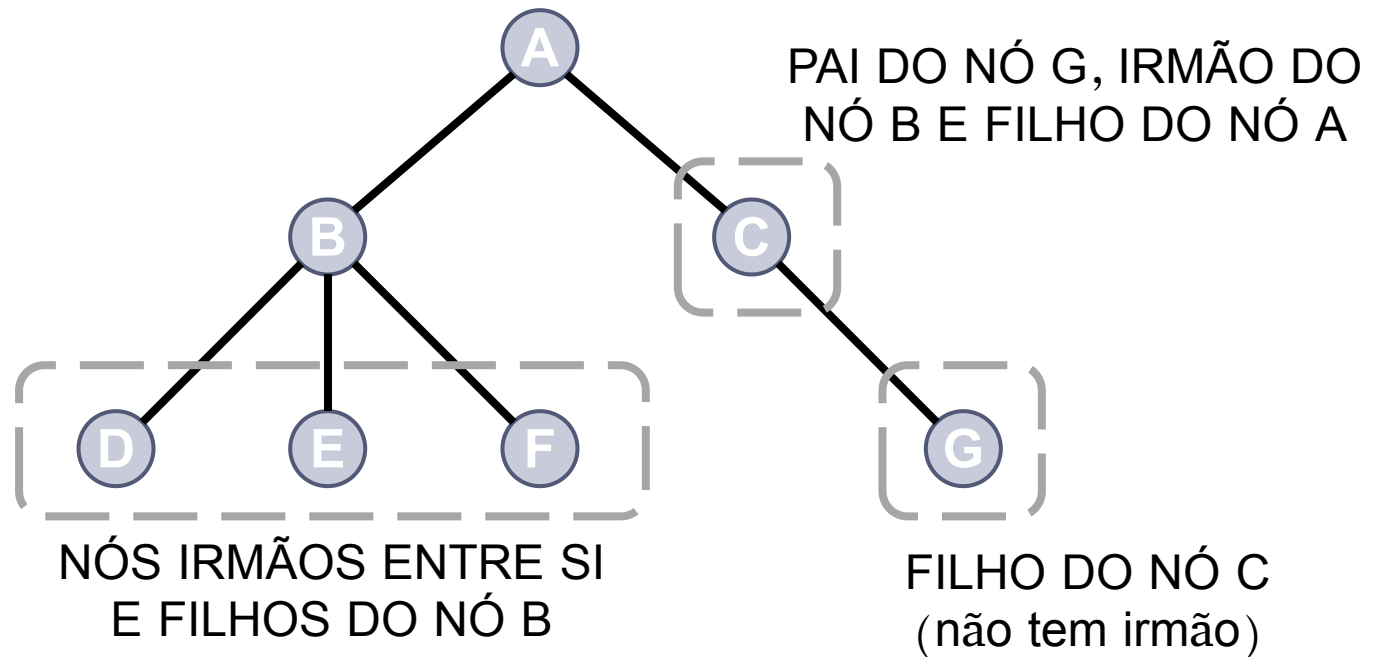
# Árvores: exemplo

---



# Árvores: exemplo

---





# Árvores: terminologias

---

**Nó raiz:** nó localizado no **topo da árvore**

**Não possui nó pai**

Contém 0 ou + subárvores, cujas raízes são seus nós filhos

**OBS: nem toda estrutura árvore possui nó raiz**

# Árvores: terminologias

---

**Nó raiz:** nó localizado no **topo da árvore**

**Não possui nó pai**

Contém 0 ou + subárvores, cujas raízes são seus nós filhos

**OBS: nem toda estrutura árvore possui nó raiz**

**Nós folha:** vértices que não possuem nós filhos

Também podem ser denominados **nós terminais**

# Árvores: terminologias

---

**Nó raiz:** nó localizado no **topo da árvore**

**Não possui nó pai**

Contém 0 ou + subárvores, cujas raízes são seus nós filhos

**OBS: nem toda estrutura árvore possui nó raiz**

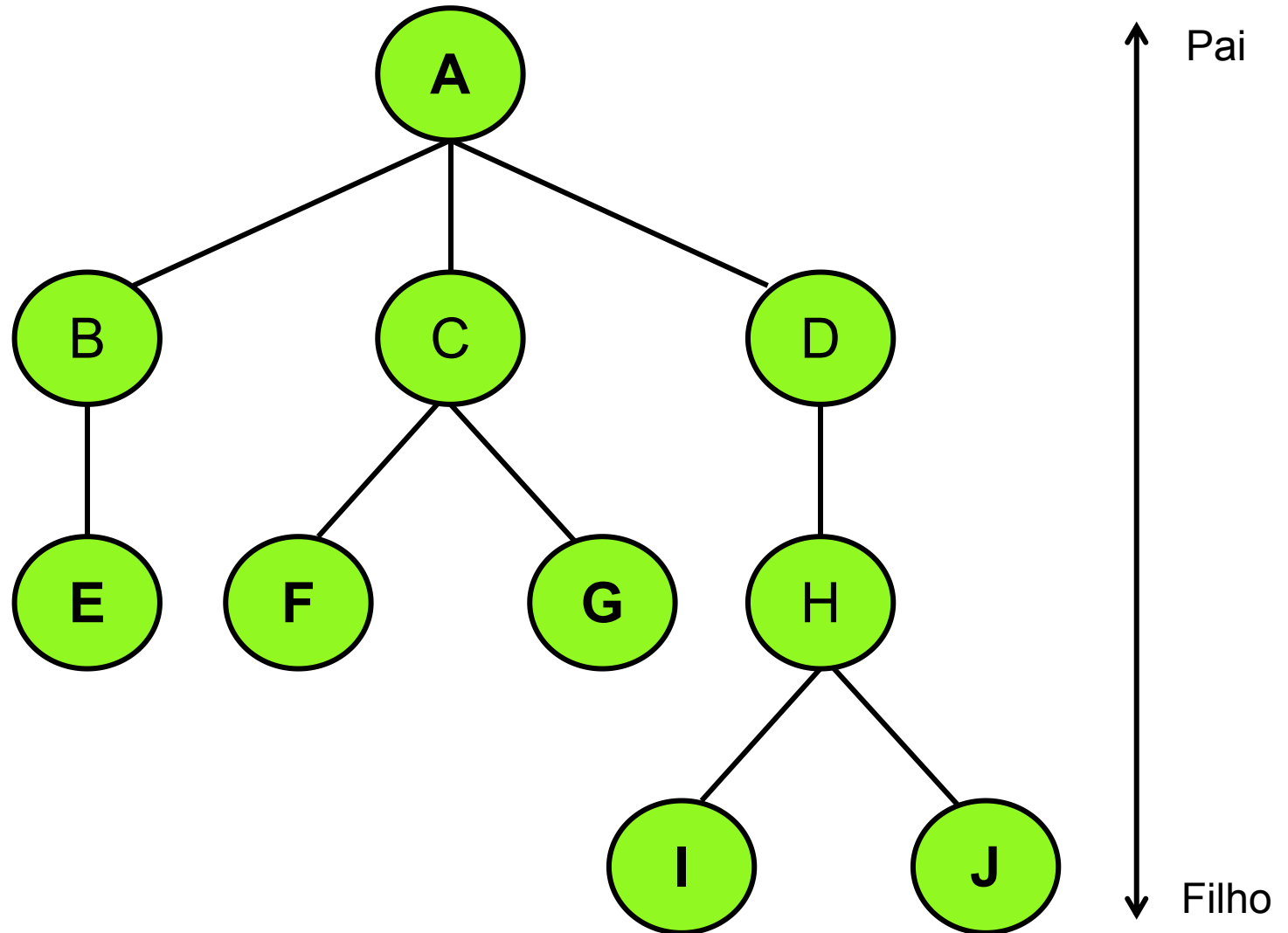
**Nós folha:** vértices que não possuem nós filhos

Também podem ser denominados **nós terminais**

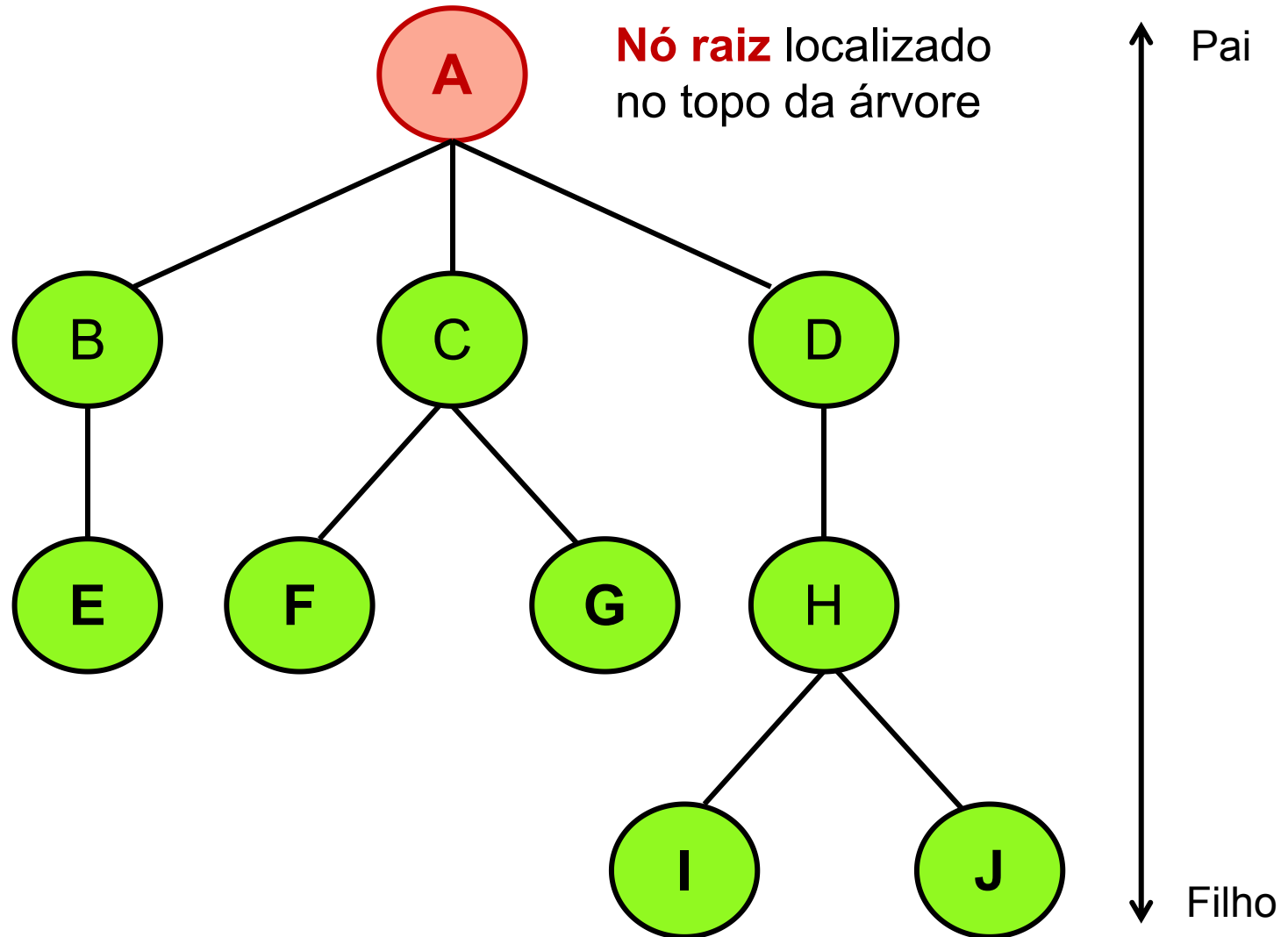
**Nós intermediários:** vértices que possuem um nó pai (não é raiz) e pelo menos um nó filho (não é terminal)

Também podem ser denominados **nós não terminais**

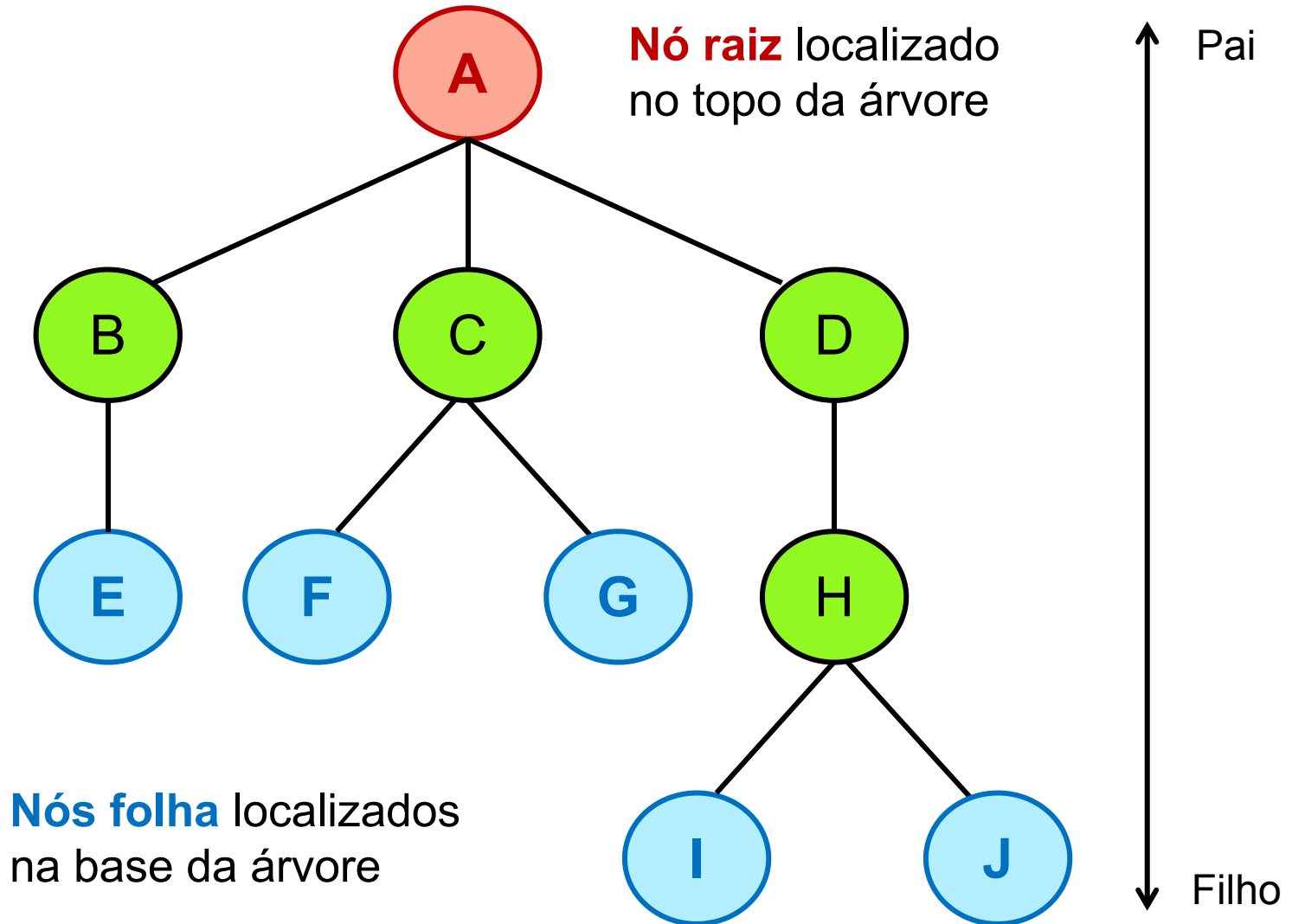
# Árvores: exemplo



# Árvores: exemplo



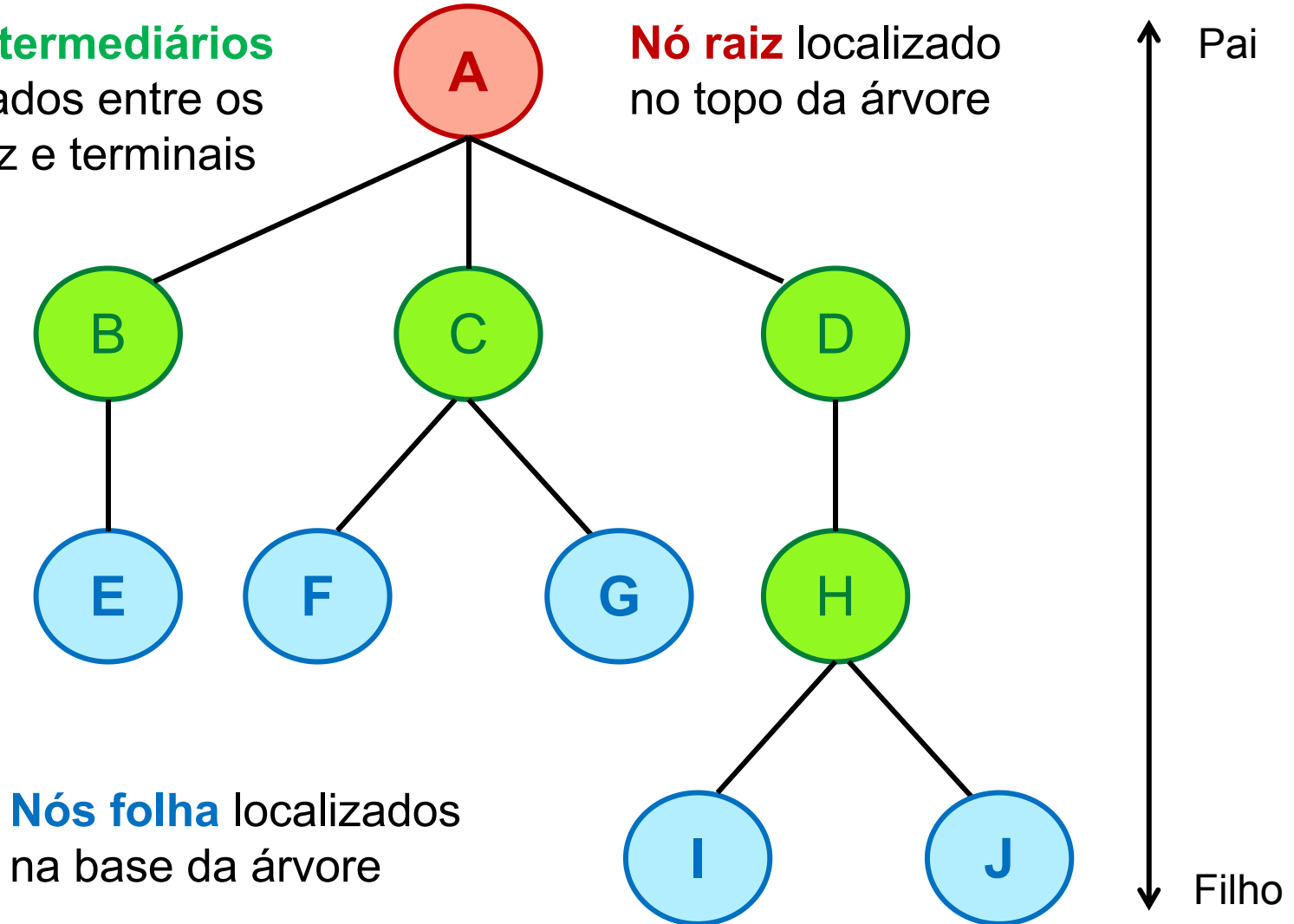
# Árvores: exemplo



# Árvores: exemplo

**Nós intermediários**  
localizados entre os  
nós raiz e terminais

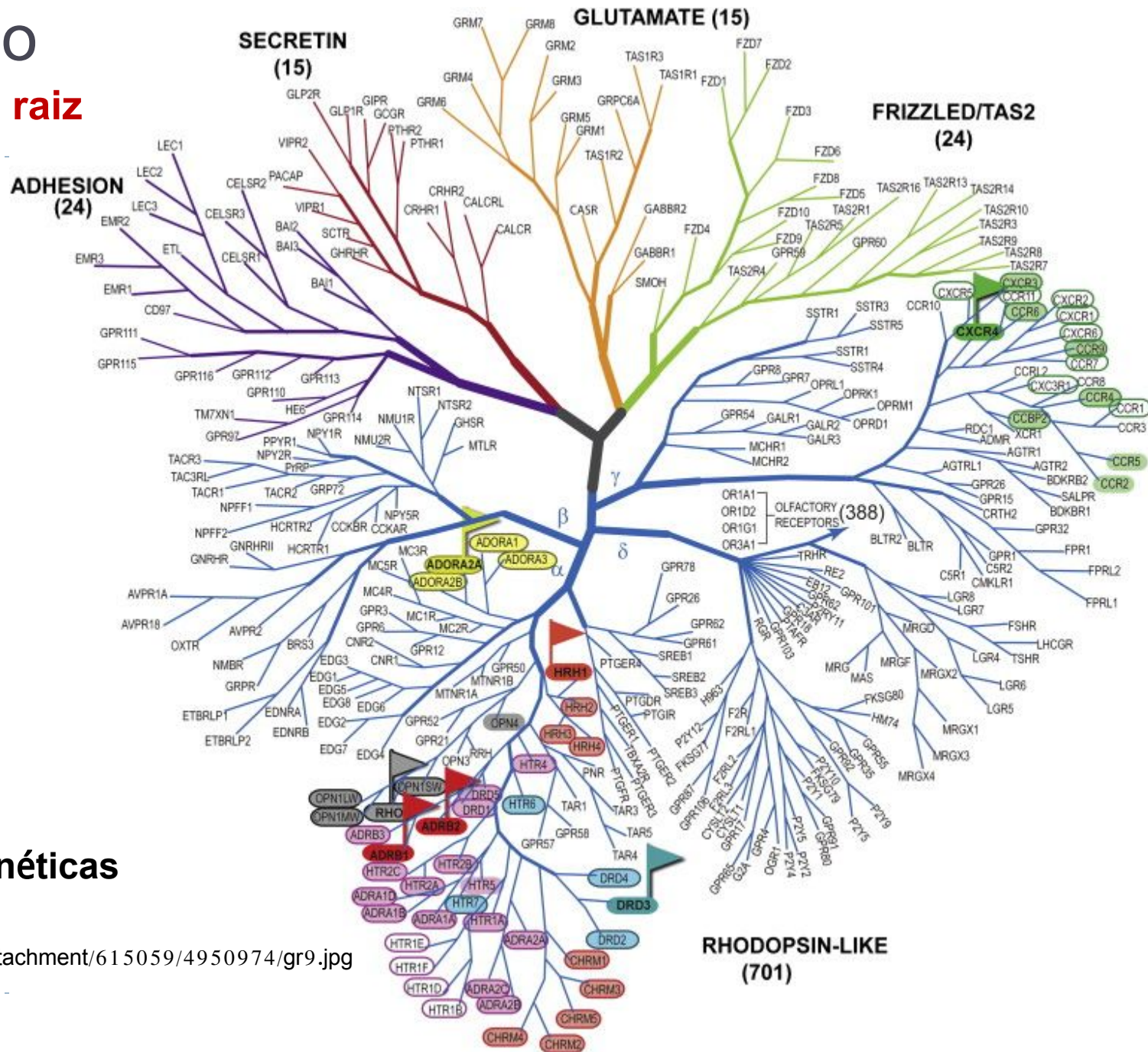
**Nó raiz** localizado  
no topo da árvore



**Nós folha** localizados  
na base da árvore

# Exemplo

## Árvore sem raiz



Cada nó é uma  
proteína

Arestas indicam  
relações filogenéticas

<http://www.cell.com/cms/attachment/615059/4950974/gr9.jpg>



# Árvores: terminologias

---

**Caminho** é a **sequência de nós** consecutivos distintos  $(n_1, n_2, \dots, n_k)$ , tal que  $n_i$  é filho de  $n_{i-1}$  e pai de  $n_{i+1}$

# Árvores: terminologias

---

**Caminho** é a **sequência de nós** consecutivos distintos  $(n_1, n_2, \dots, n_k)$ , tal que  $n_i$  é filho de  $n_{i-1}$  e pai de  $n_{i+1}$

Existe um **único caminho** entre o nó raiz e cada nó da árvore

Diferente do grafo, onde pode existir + de um caminho

# Árvores: terminologias

---

**Caminho** é a **sequência de nós** consecutivos distintos  $(n_1, n_2, \dots, n_k)$ , tal que  $n_i$  é filho de  $n_{i-1}$  e pai de  $n_{i+1}$

Existe um **único caminho** entre o nó raiz e cada nó da árvore

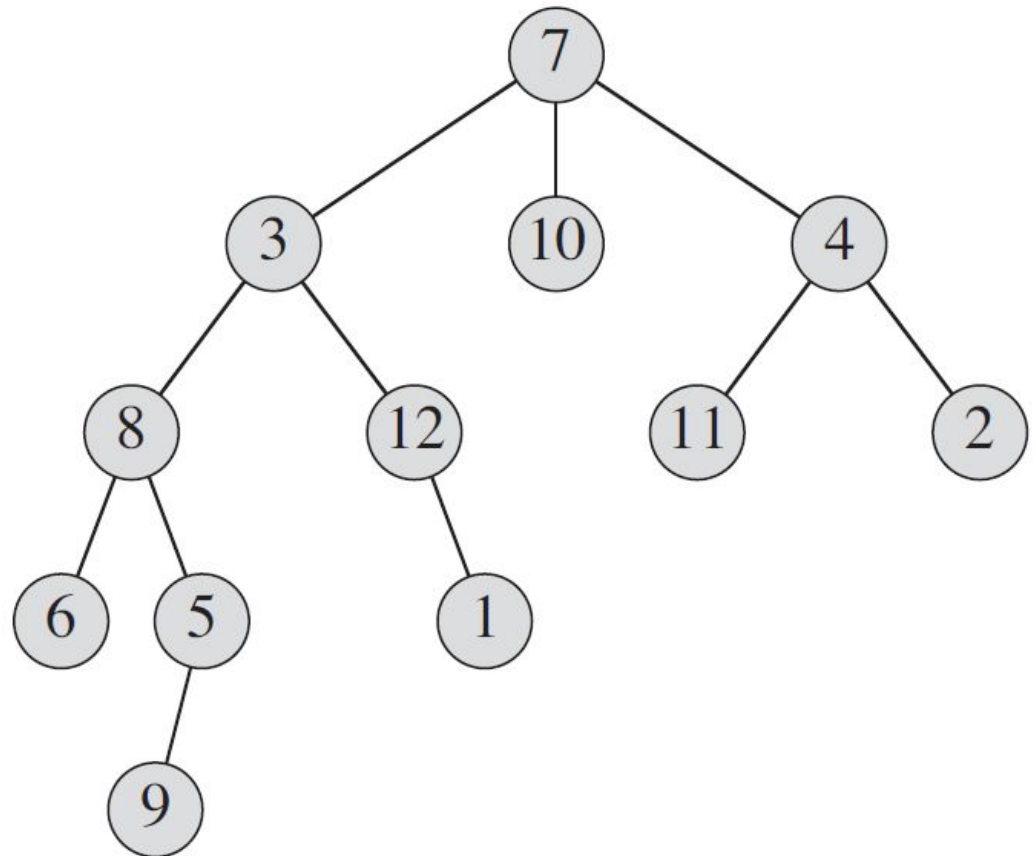
Diferente do grafo, onde pode existir + de um caminho

O **comprimento de um caminho** é o **número de arestas** do caminho (ou **número de vértices - 1**)

Caminho de comprimento ZERO é do nó para ele mesmo

# Árvores: exemplo

*Comprimento* = N° de arestas = N° de vértices - 1



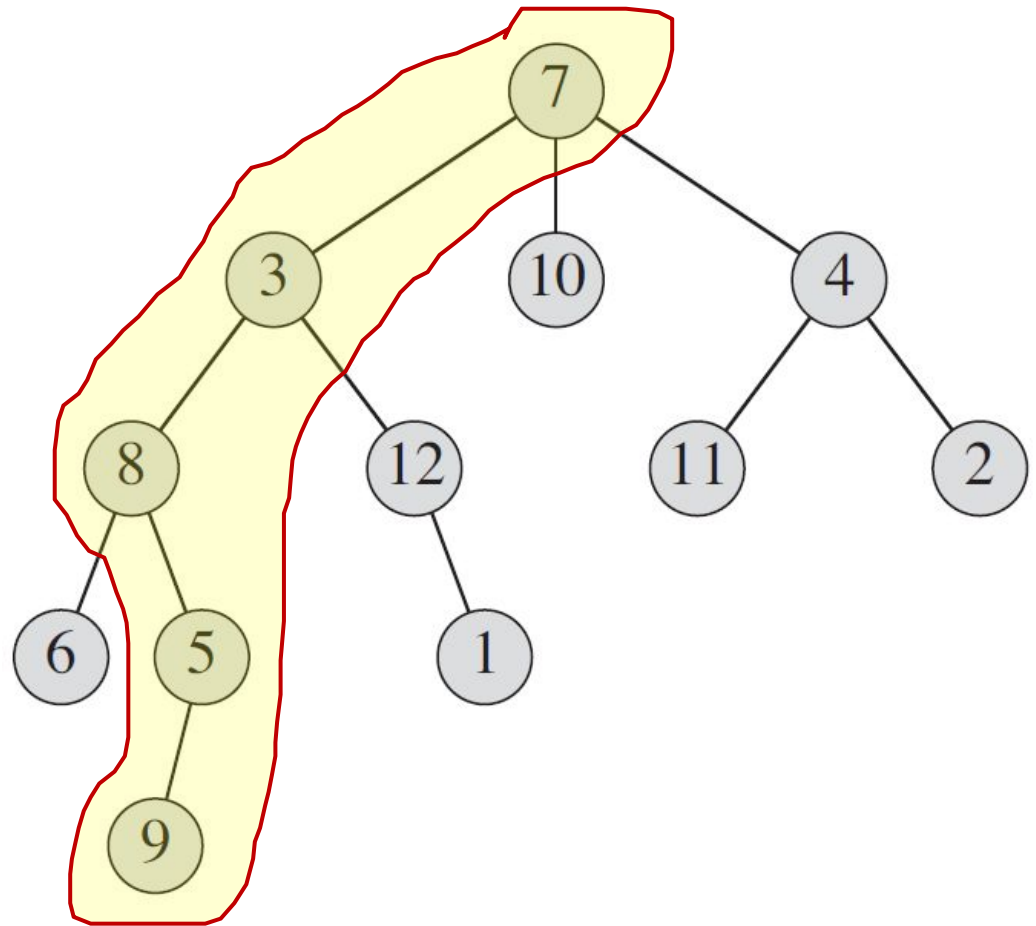
# Árvores: exemplo

*Comprimento* = N° de arestas = N° de vértices - 1

**Caminho de 7 a 9:**

**(7, 3, 8, 5, 9)**

**Comprimento = 4**



# Árvores: exemplo

*Comprimento* = N° de arestas = N° de vértices - 1

Caminho de 7 a 9:

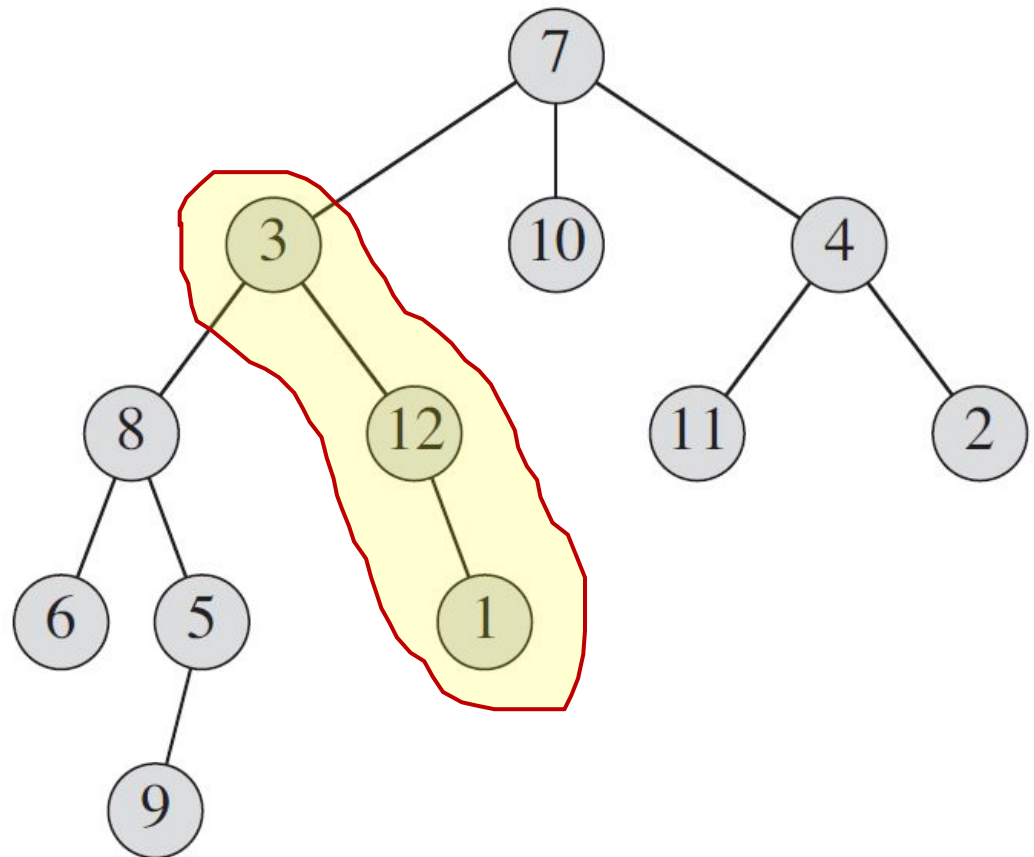
(7, 3, 8, 5, 9)

Comprimento = 4

**Caminho de 3 a 1:**

(3, 12, 1)

**Comprimento = 2**



# Árvores: terminologias

---

**Ancestral:**  $n_i$  é dito ancestral de  $n_k$  se existe um caminho de  $n_i$  para  $n_k$

$n_i$  está acima de  $n_k$

O nó raiz é ancestral de todos os demais nós

# Árvores: terminologias

---

**Ancestral:**  $n_i$  é dito ancestral de  $n_k$  se existe um caminho de  $n_i$  para  $n_k$

$n_i$  está acima de  $n_k$

O nó raiz é ancestral de todos os demais nós

**Descendente:**  $n_i$  é dito descendente de  $n_k$  se existe um caminho que chega em  $n_i$  a partir de  $n_k$

$n_i$  está abaixo de  $n_k$

Nós terminais (folha) não tem descendentes



# Árvores: terminologias

---

**Altura de um nó** é o **maior caminho** entre ele e seus descendentes que sejam nó folha

**Altura de uma árvore = altura do nó raiz**

**Nível ou profundidade de um nó** é o comprimento do caminho entre o nó raiz e ele

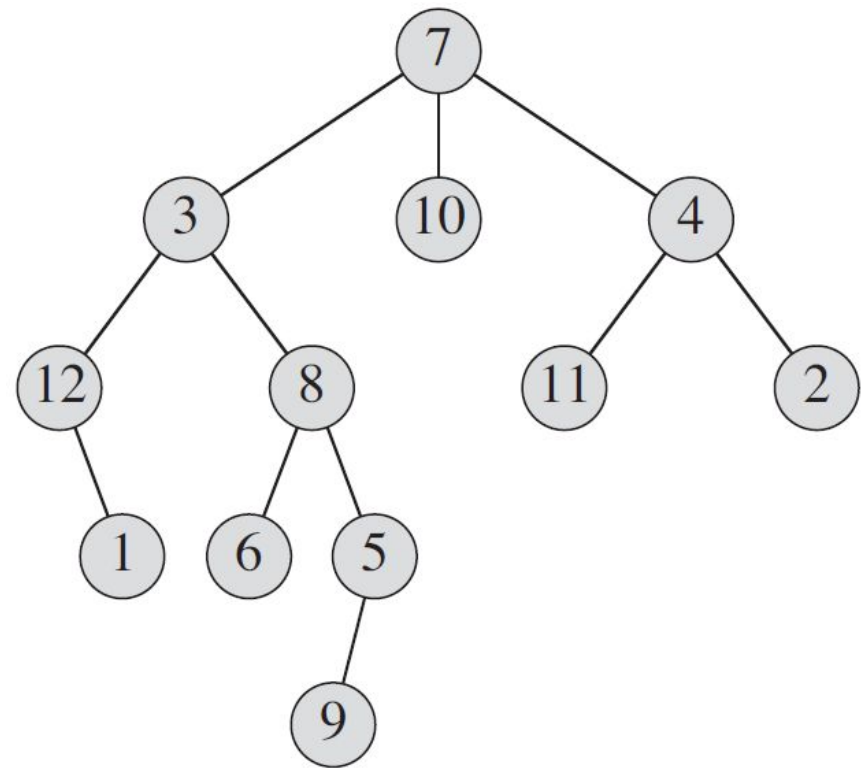
Nível (profundidade) do nó raiz é **ZERO**

Se um nó tem nível  $i$ , seus filhos tem nível  $i+1$

**Profundidade de uma árvore = sua altura**

# Árvores: exemplo

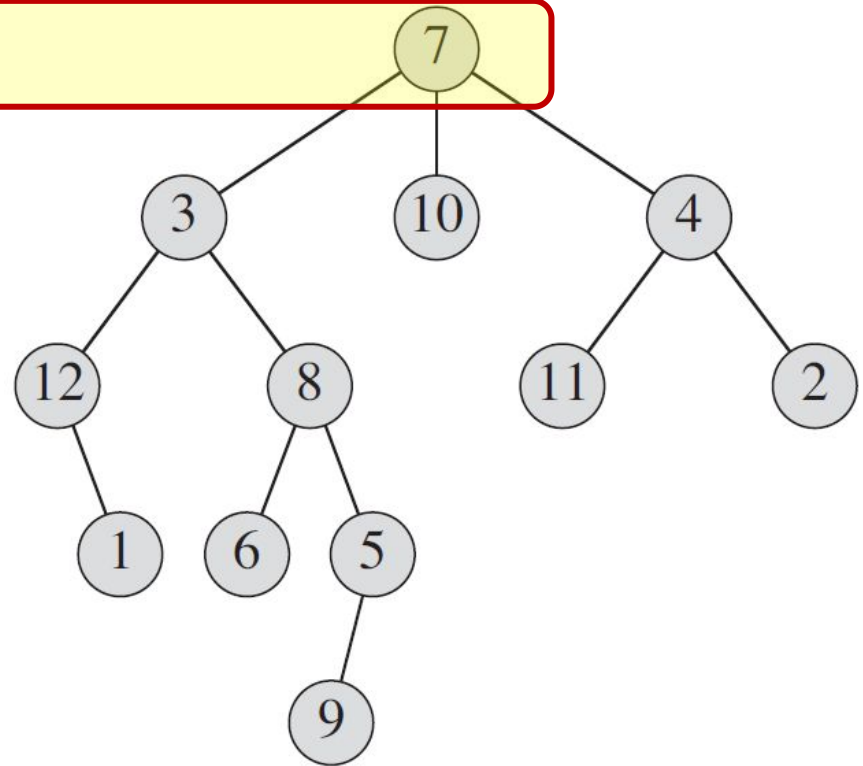
A **profundidade** ou **nível** de um nó é o comprimento do caminho da raiz até o nó (**nível do pai + 1**)



# Árvores: exemplo

A **profundidade** ou **nível** de um nó é o comprimento do caminho da raiz até o nó (**nível do pai + 1**)

Nível 0 (nó 7)

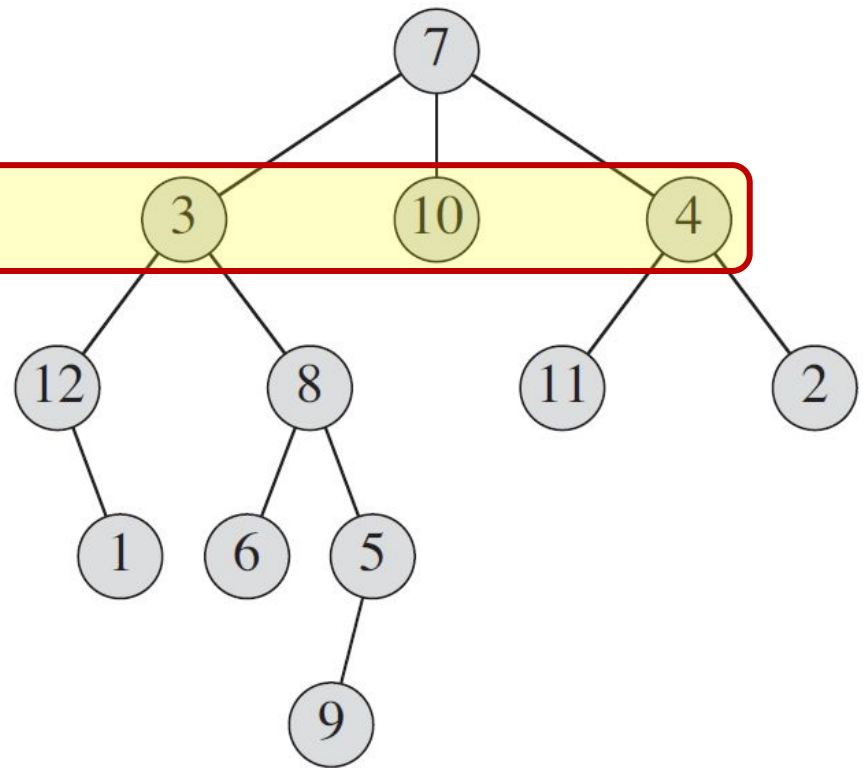


# Árvores: exemplo

A **profundidade** ou **nível** de um nó é o comprimento do caminho da raiz até o nó (**nível do pai + 1**)

Nível 0 (nó 7)

Nível 1 (nós 3, 10 e 4)



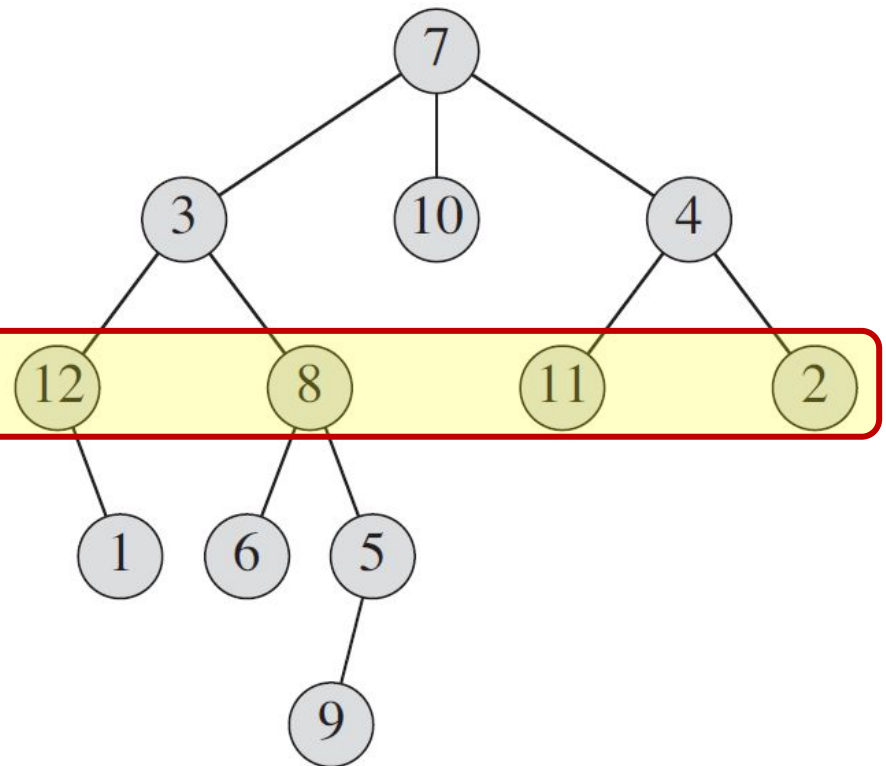
# Árvores: exemplo

A **profundidade** ou **nível** de um nó é o comprimento do caminho da raiz até o nó (**nível do pai + 1**)

Nível 0 (nó 7)

Nível 1 (nós 3, 10 e 4)

Nível 2 (nós 12, 8, 11 e 2)



# Árvores: exemplo

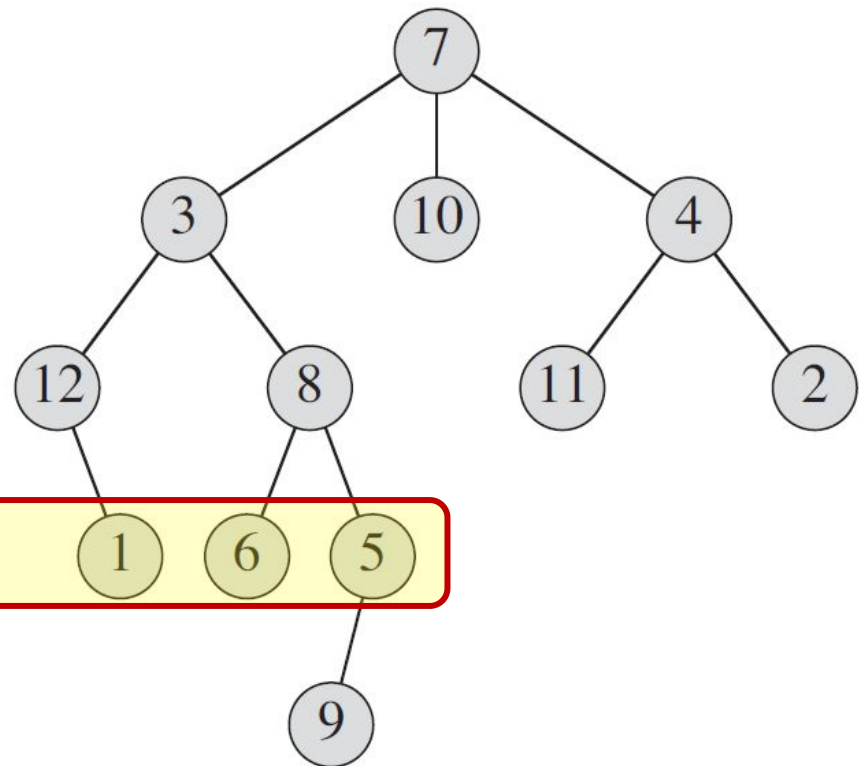
A **profundidade ou nível** de um nó é o comprimento do caminho da raiz até o nó (**nível do pai + 1**)

Nível 0 (nó 7)

Nível 1 (nós 3, 10 e 4)

Nível 2 (nós 12, 8, 11 e 2)

Nível 3 (nós 1, 6 e 5)



# Árvores: exemplo

A **profundidade** ou **nível** de um nó é o comprimento do caminho da raiz até o nó (**nível do pai + 1**)

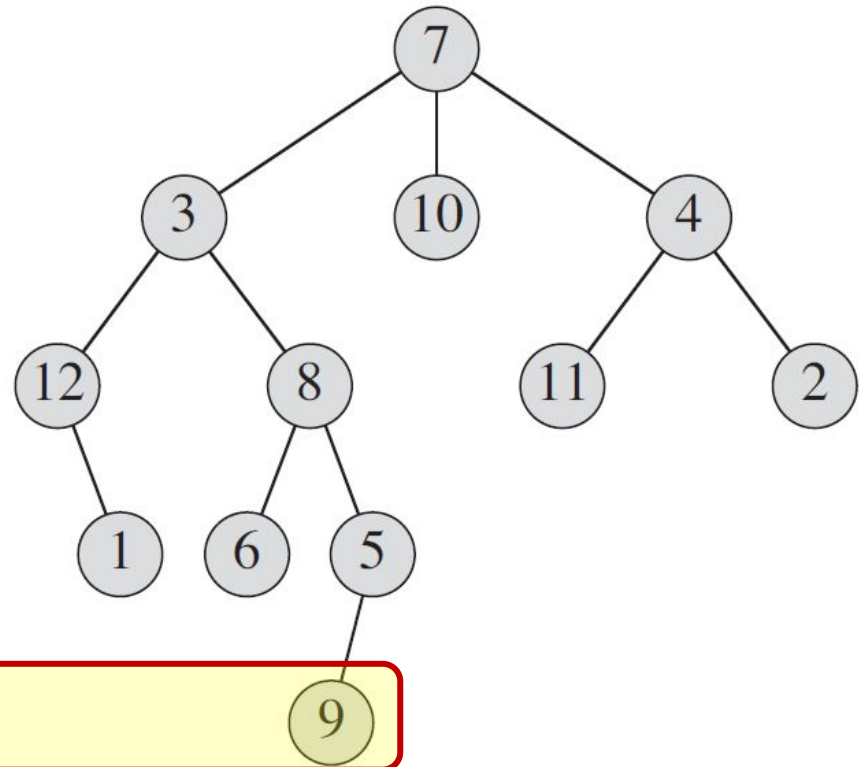
Nível 0 (nó 7)

Nível 1 (nós 3, 10 e 4)

Nível 2 (nós 12, 8, 11 e 2)

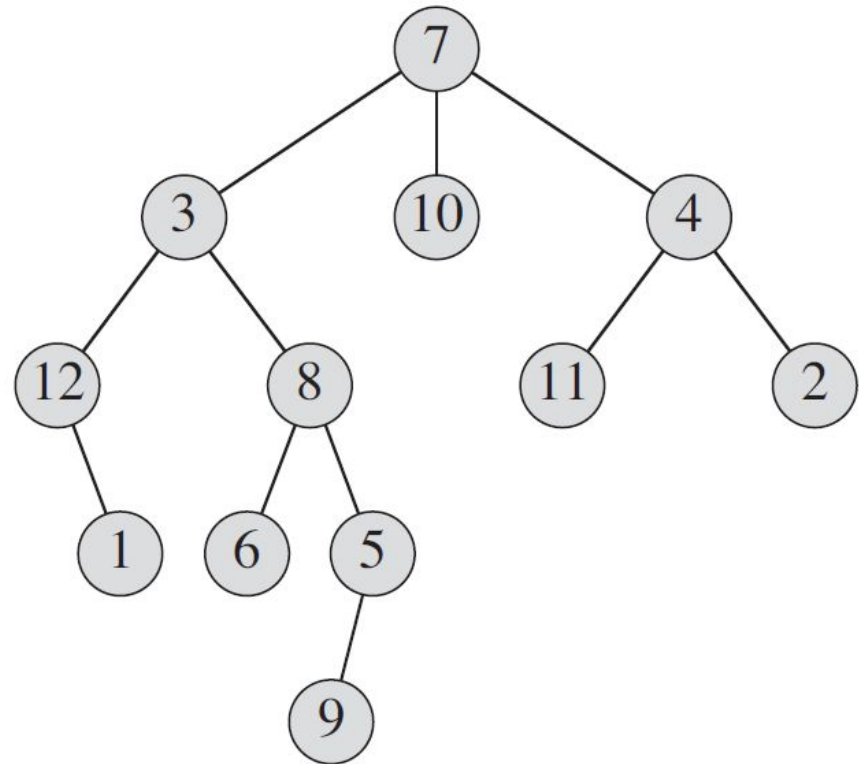
Nível 3 (nós 1, 6 e 5)

Nível 4 (nó 9)



# Árvores: exemplo

A **altura de um nó** é o **maior caminho** do nó até um nó folha

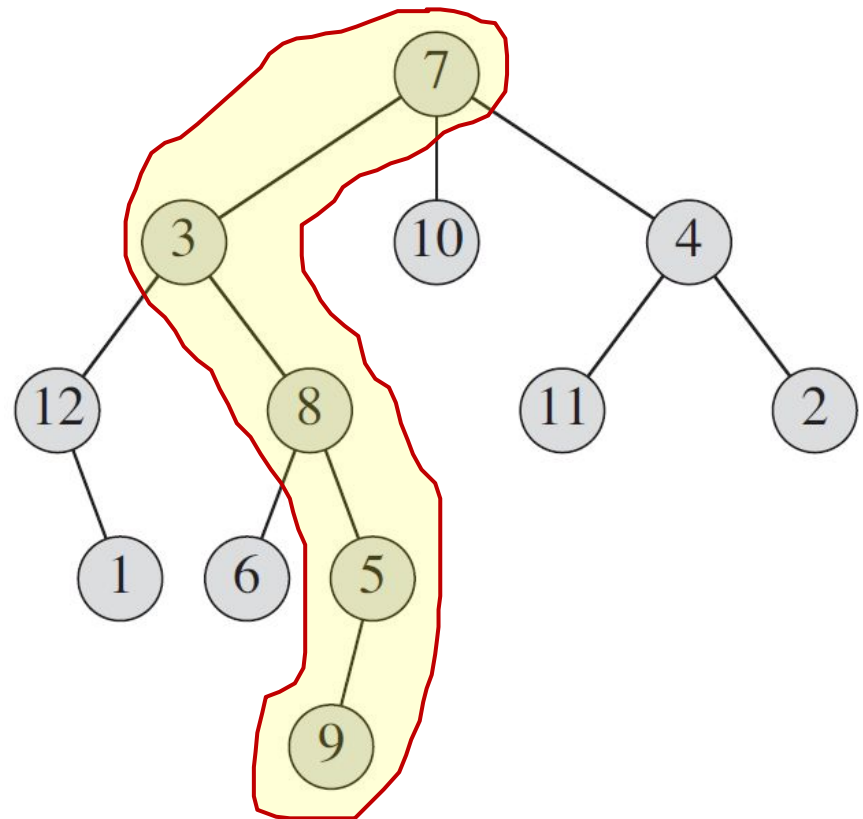




# Árvores: exemplo

A **altura de um nó** é o **maior caminho** do nó até um nó folha

Altura do **nó 7 (raíz)** = 4

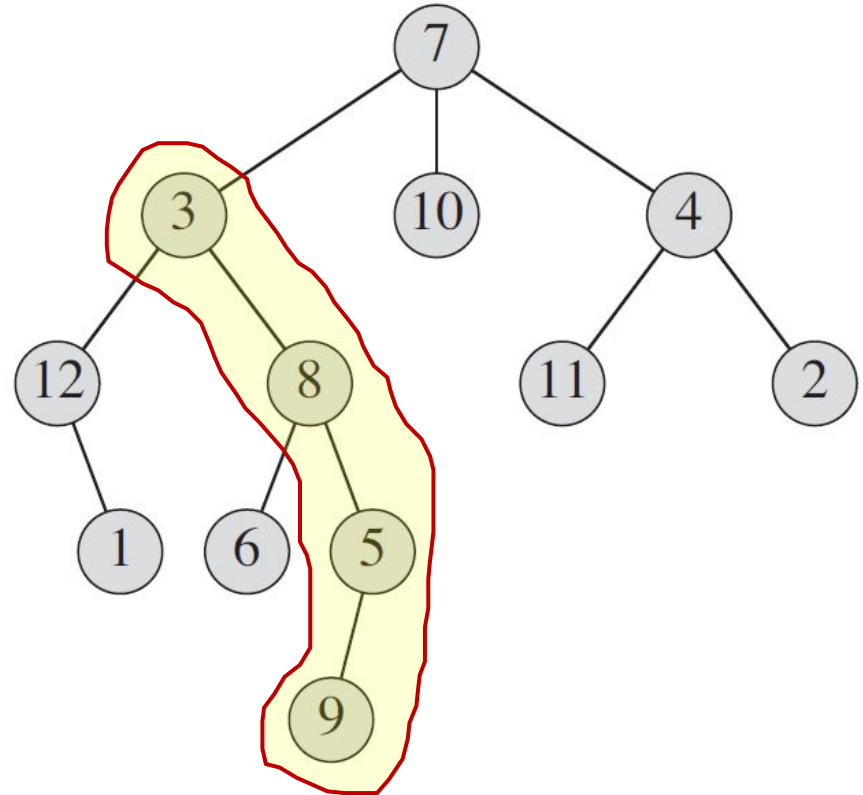


# Árvores: exemplo

A **altura de um nó** é o **maior caminho** do nó até um nó folha

Altura do nó 7 (raíz) = 4

Altura do **nó 3** = 3



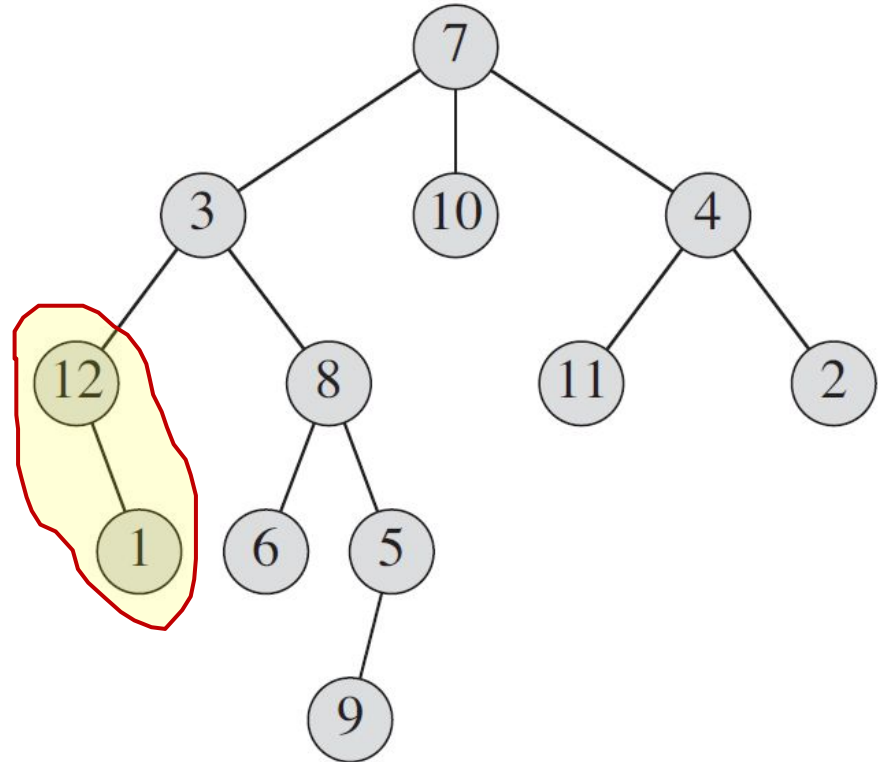
# Árvores: exemplo

A **altura de um nó** é o **maior caminho** do nó até um nó folha

Altura do nó 7 (raíz) = 4

Altura do nó 3 = 3

Altura do **nó 12** = 1



# Árvores: exemplo

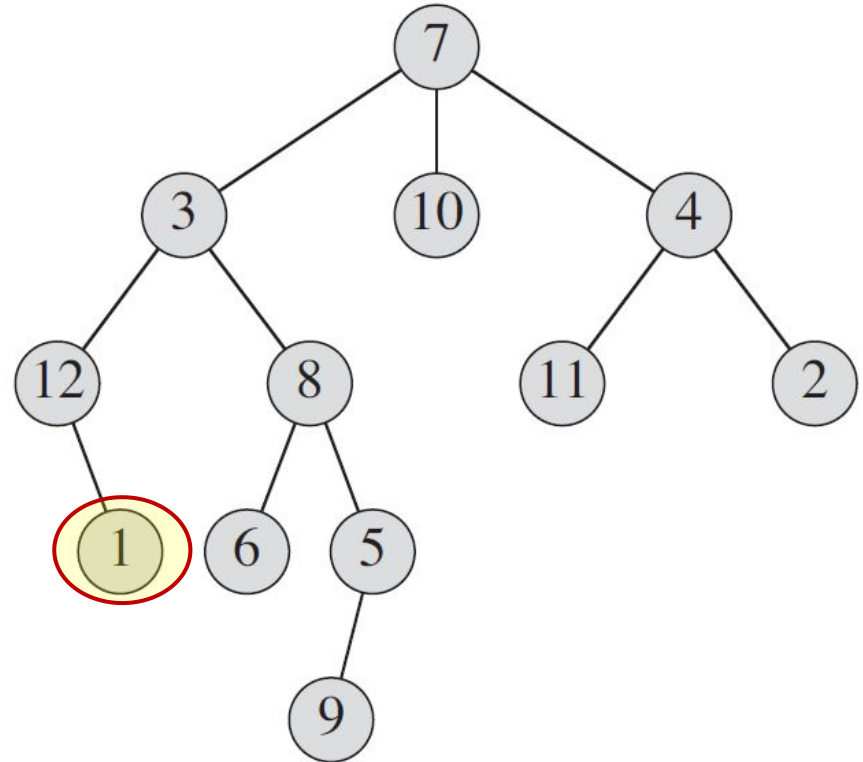
A **altura de um nó** é o **maior caminho** do nó até um nó folha

Altura do nó 7 (raíz) = 4

Altura do nó 3 = 3

Altura do nó 12 = 1

Altura do **nó 1 (folha)** = 0



# Árvores: exemplo

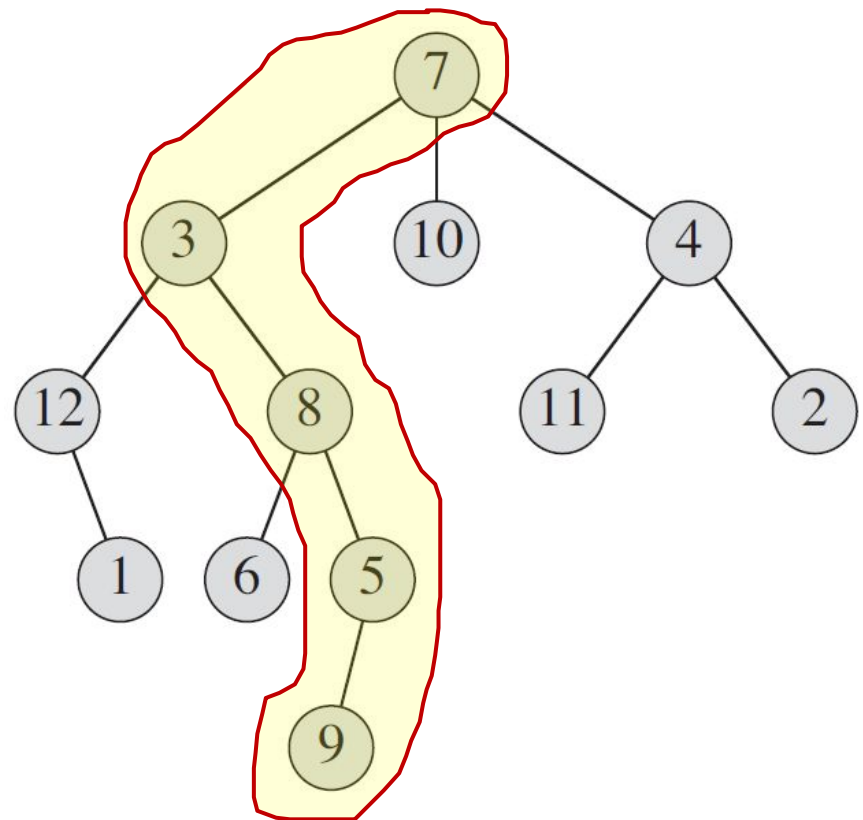
A **altura de um nó** é o **maior caminho** do nó até um nó folha

Altura do nó 7 (raíz) = 4

Altura do nó 3 = 3

Altura do nó 12 = 1

Altura do nó 1 (folha) = 0



A **altura da árvore** é a altura do nó raiz (**altura = 4**)

# Subárvores

---

**Subárvore** é uma estrutura disjunta de árvore que representa um subconjunto da hierarquia da árvore original

Formada por um nó da árvore e **todos os seus descendentes**

Um mesmo nó não pode aparecer em mais de uma subárvore

# Subárvores

---

**Subárvore** é uma estrutura disjunta de árvore que representa um subconjunto da hierarquia da árvore original

Formada por um nó da árvore e **todos os seus descendentes**

Um mesmo nó não pode aparecer em mais de uma subárvore

Não pode ser gerada pela remoção de nós subordinados

**Ex:** nós folha

# Subárvores

---

**Subárvore** é uma estrutura disjunta de árvore que representa um subconjunto da hierarquia da árvore original

Formada por um nó da árvore e **todos os seus descendentes**

Um mesmo nó não pode aparecer em mais de uma subárvore

Não pode ser gerada pela remoção de nós subordinados

**Ex:** nós folha

Como a árvore original, **possui nó raiz e nós terminais**



# Subárvores

---

**Subárvore** é uma estrutura disjunta de árvore que representa um subconjunto da hierarquia da árvore original

Formada por um nó da árvore e **todos os seus descendentes**

Um mesmo nó não pode aparecer em mais de uma subárvore

Não pode ser gerada pela remoção de nós subordinados

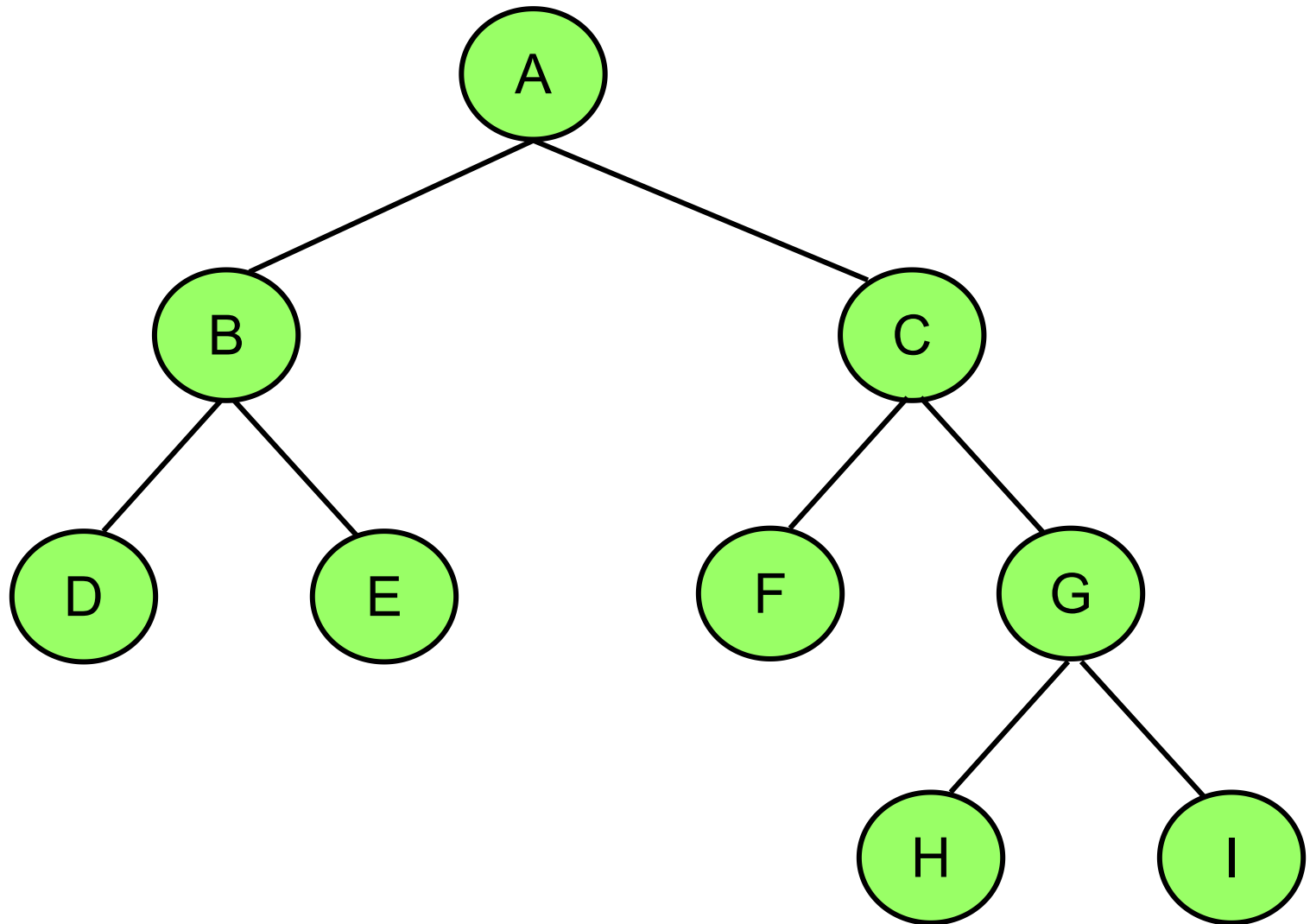
**Ex:** nós folha

Como a árvore original, **possui nó raiz e nós terminais**

Raízes das subárvores podem ter filhos que são raízes de outras subárvores (**definição recursiva**)

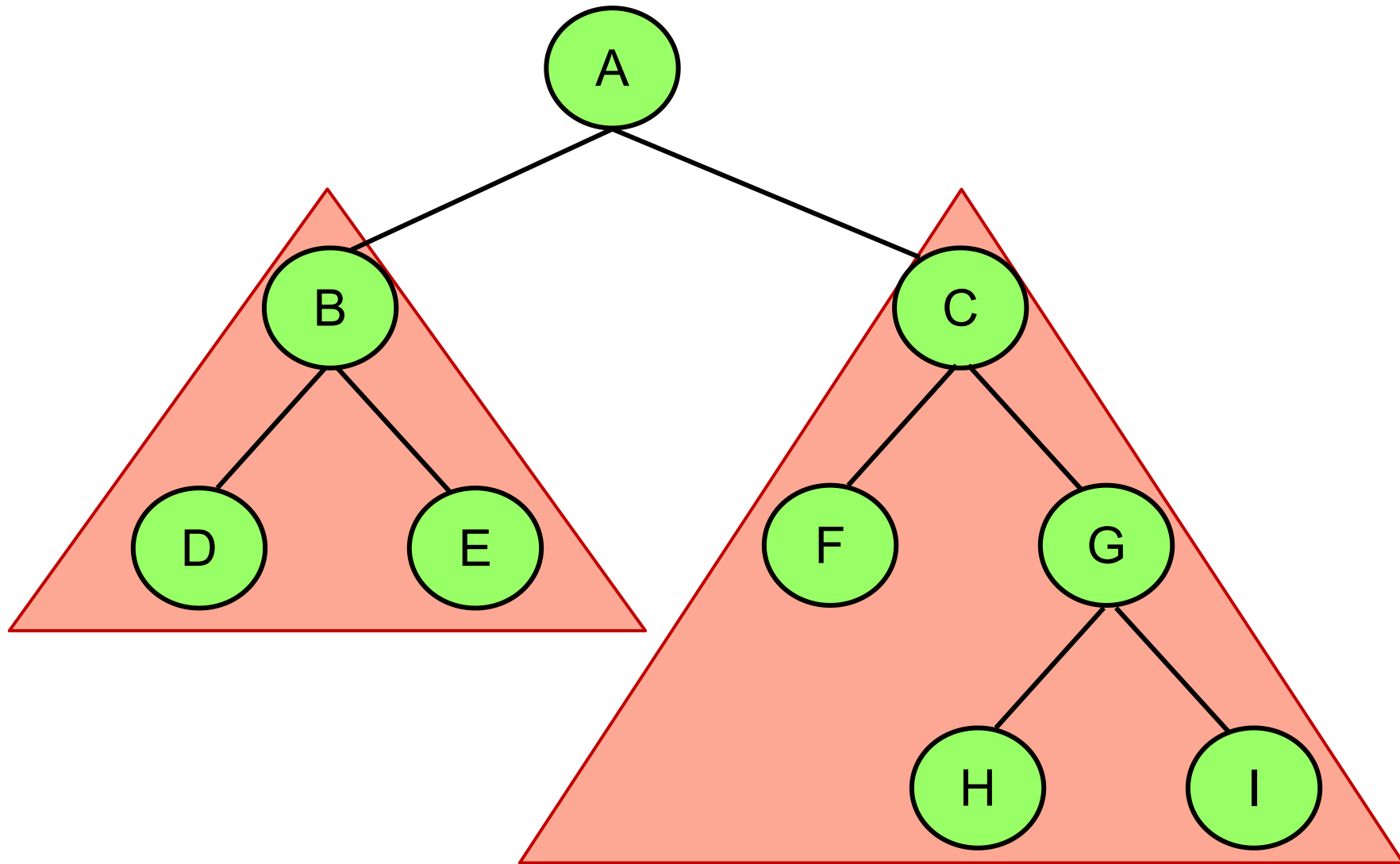
# Exemplo subárvores

---

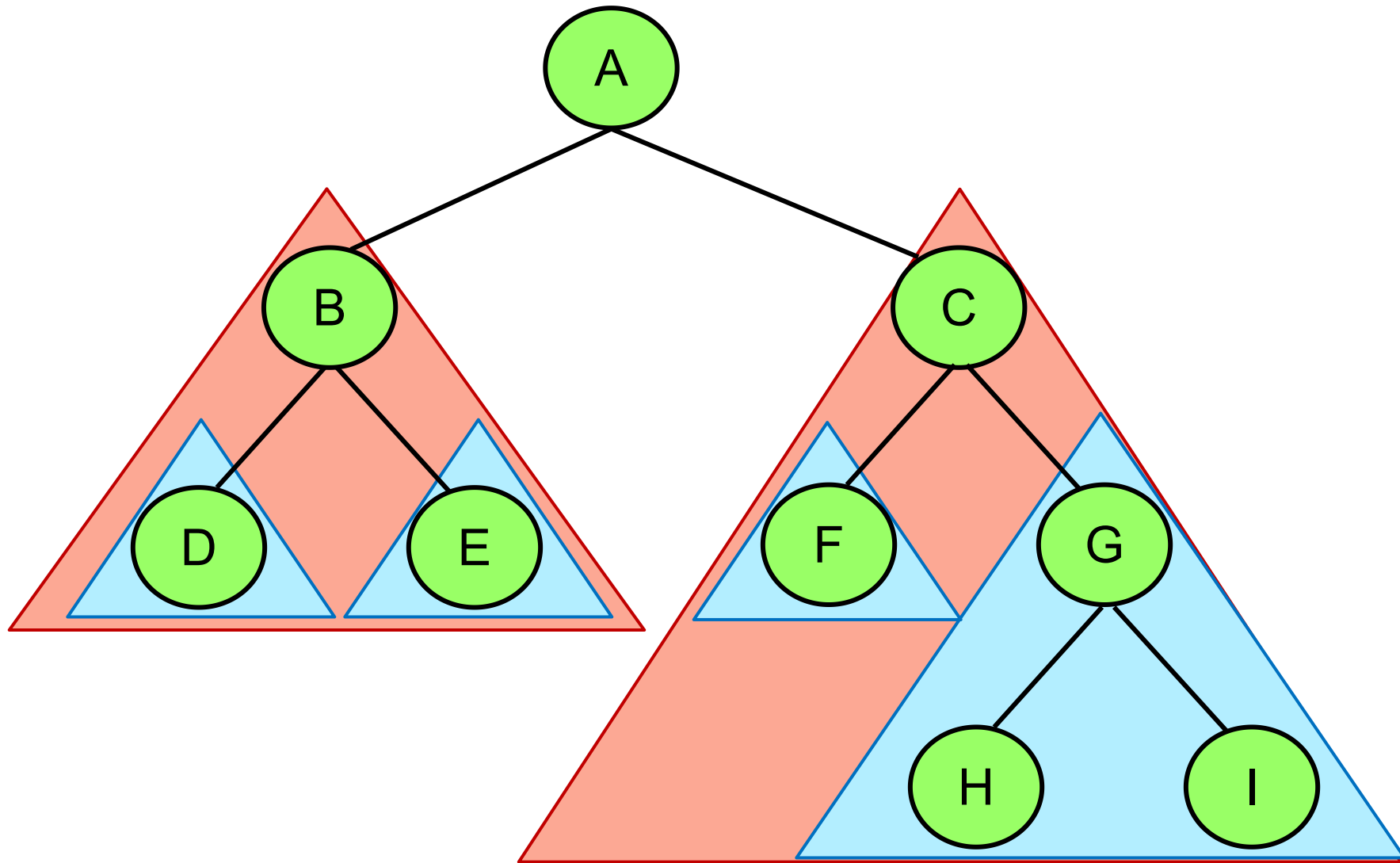


# Exemplo subárvores

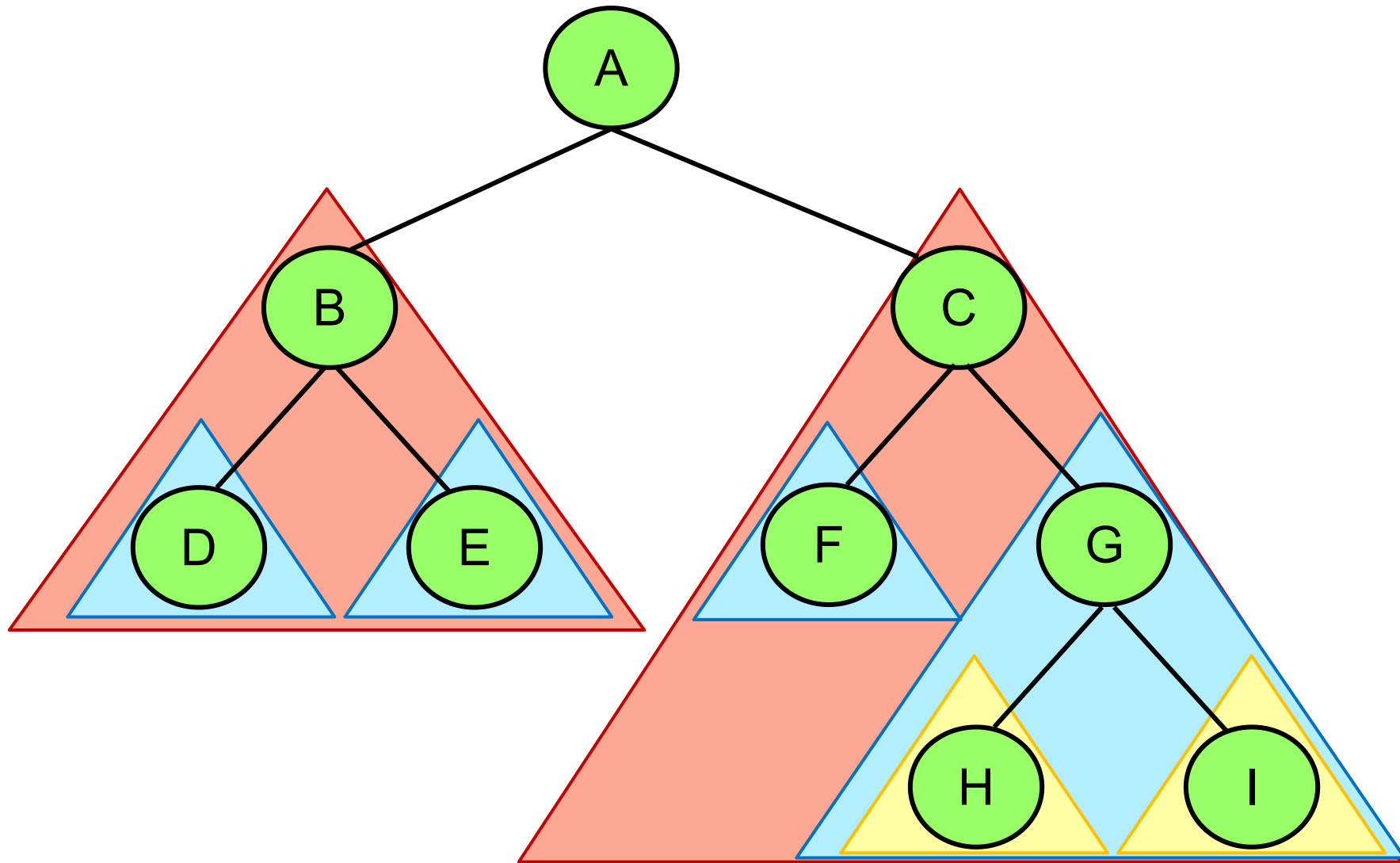
---



# Exemplo subárvores



# Exemplo subárvores



# Árvores: definição recursiva

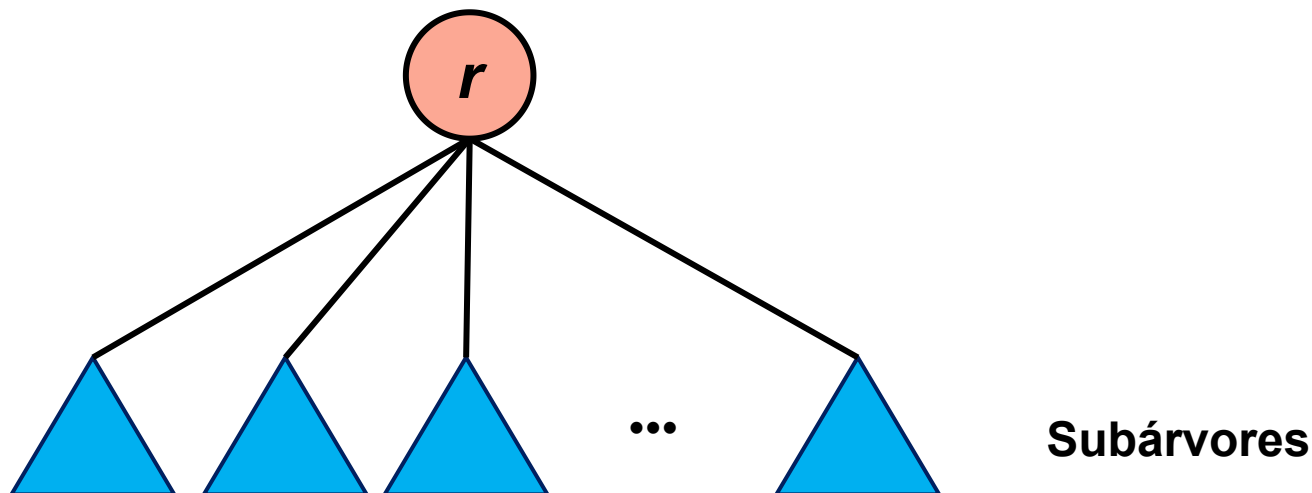
Seja **A** uma árvore. Então **A** é:

Uma **árvore nula**: árvore sem nós (**vazia**)

Uma árvore formada por **um nó raiz  $r$**  e **0 ou + subárvores**

Os nós raízes das subárvores são os filhos de  $r$

Cada **subárvore** é uma árvore



# Árvores: terminologias

---

**Grau de um nó** é o número de suas subárvores filhas

Indica o **número de filhos do nó**

# Árvores: terminologias

---

**Grau de um nó** é o número de suas subárvores filhas

Indica o **número de filhos do nó**

**Nós folha** tem **grau ZERO**

Não são consideradas as **subárvores nulas**



# Árvores: terminologias

---

**Grau de um nó** é o número de suas subárvores filhas

Indica o **número de filhos do nó**

**Nós folha** tem **grau ZERO**

Não são consideradas as **subárvores nulas**

**Nós de derivação** são aqueles com **grau maior que ZERO**

# Árvores: terminologias

---

**Grau de um nó** é o número de suas subárvores filhas

Indica o **número de filhos do nó**

**Nós folha** tem **grau ZERO**

Não são consideradas as **subárvores nulas**

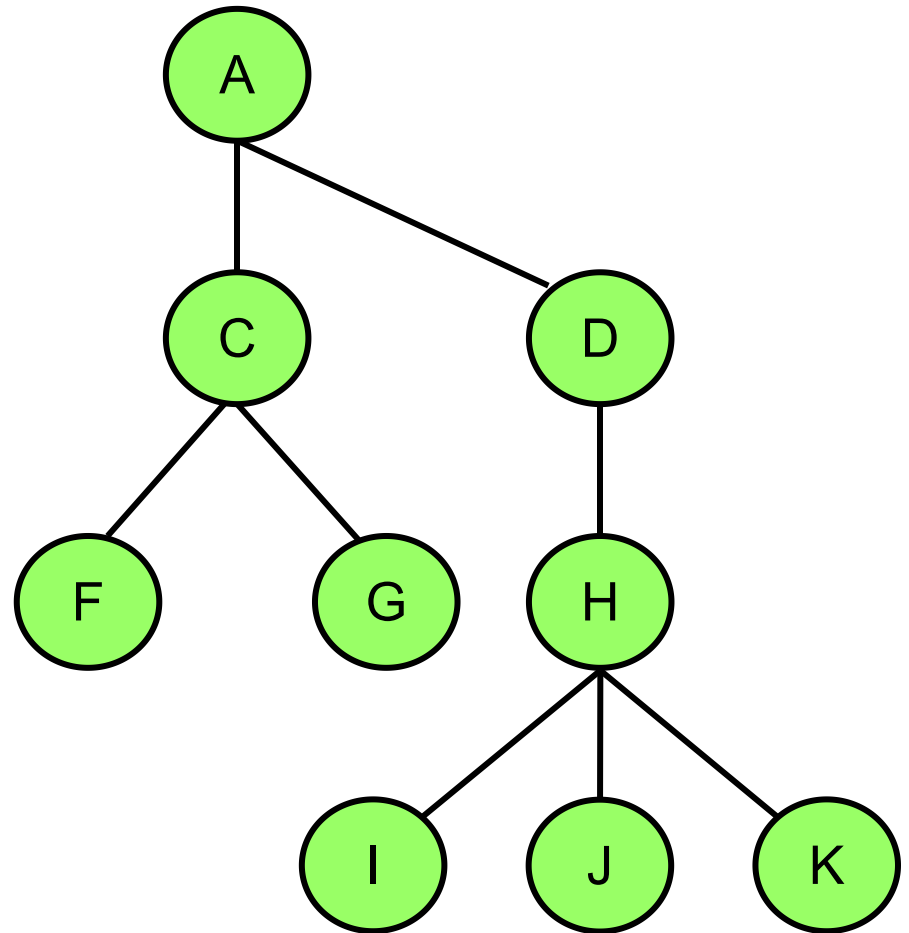
**Nós de derivação** são aqueles com **grau maior que ZERO**

**Grau da árvore** é o grau do nó com mais filhos

Indica o **maior grau** entre os seus nós

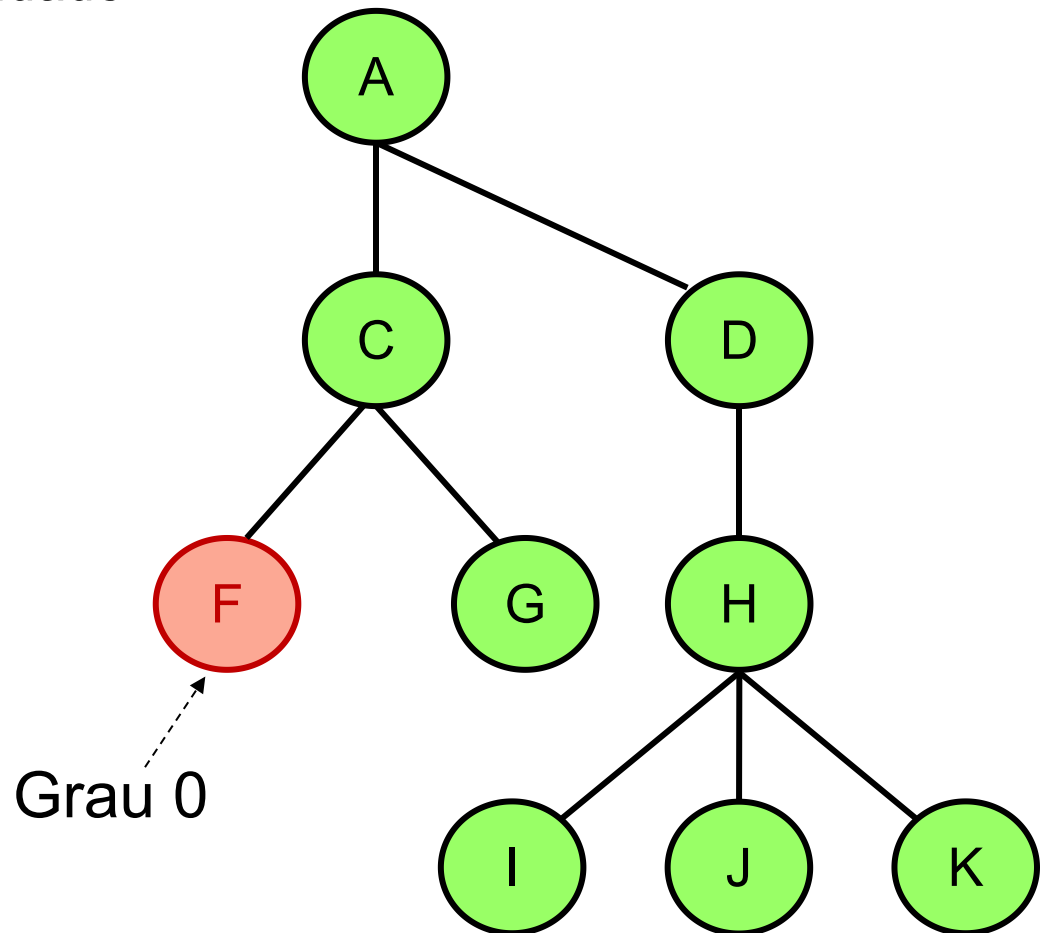
# Árvores: exemplo

O grau de um nó é a quantidade de filhos do nó



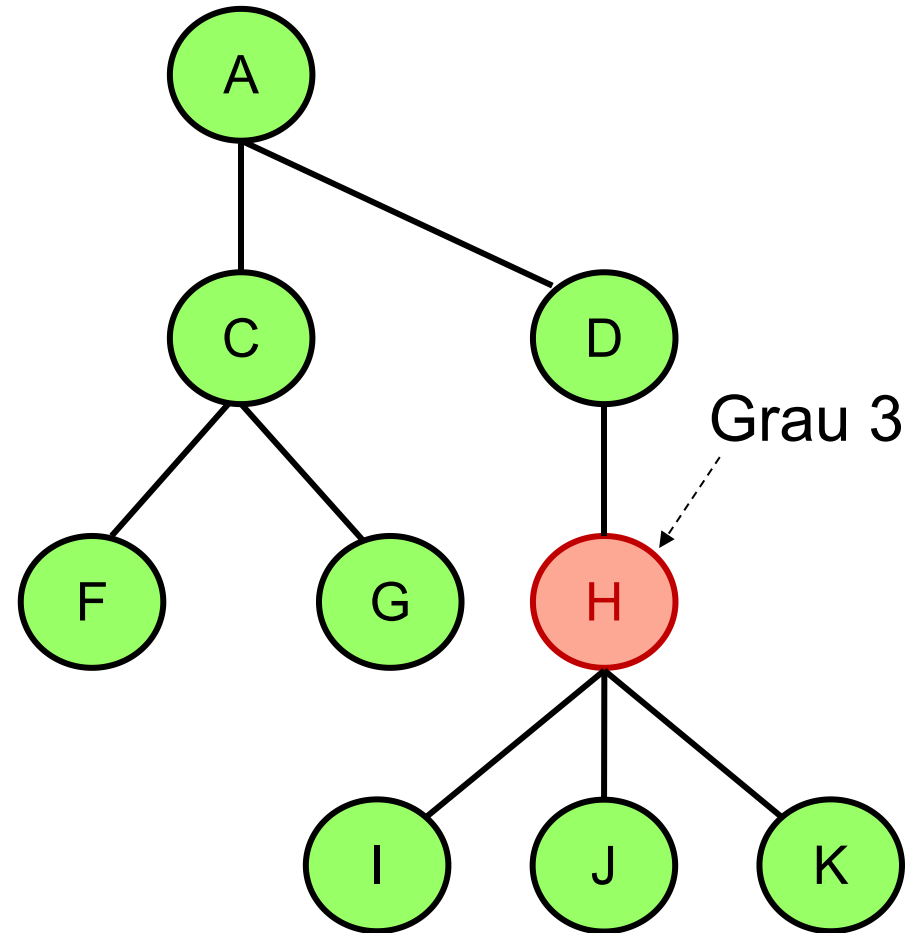
# Árvores: exemplo

O grau de um nó é a quantidade de filhos do nó



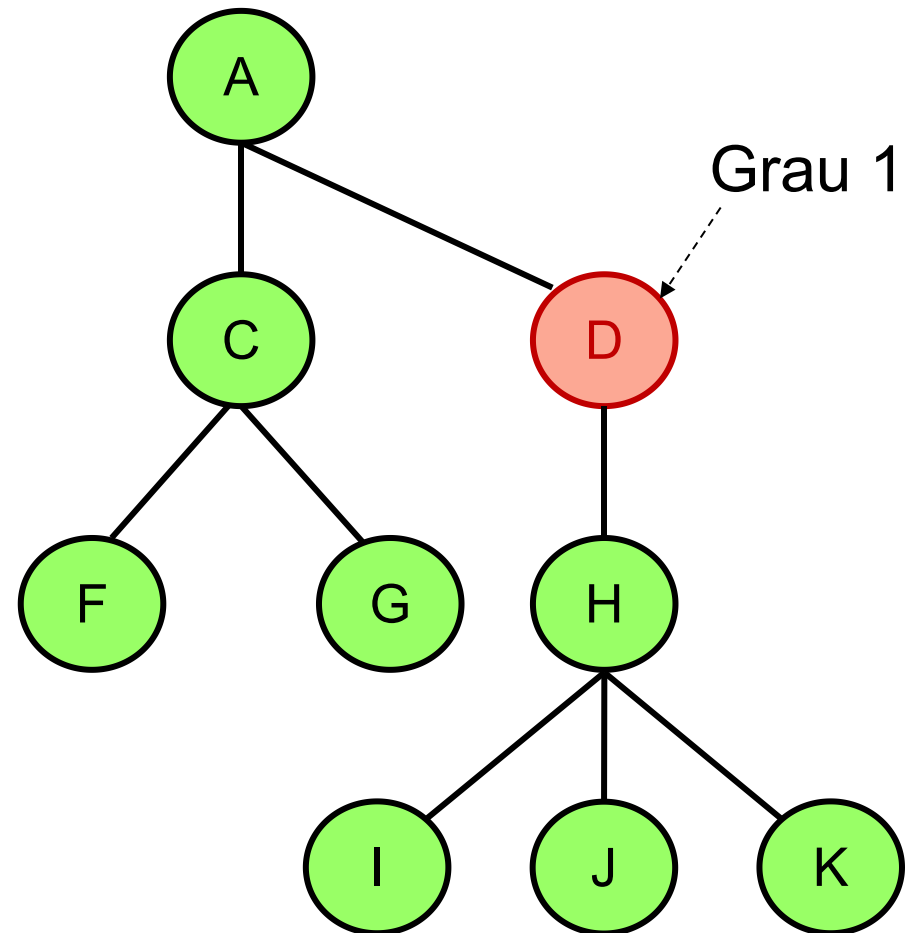
# Árvores: exemplo

O grau de um nó é a quantidade de filhos do nó



# Árvores: exemplo

O grau de um nó é a quantidade de filhos do nó

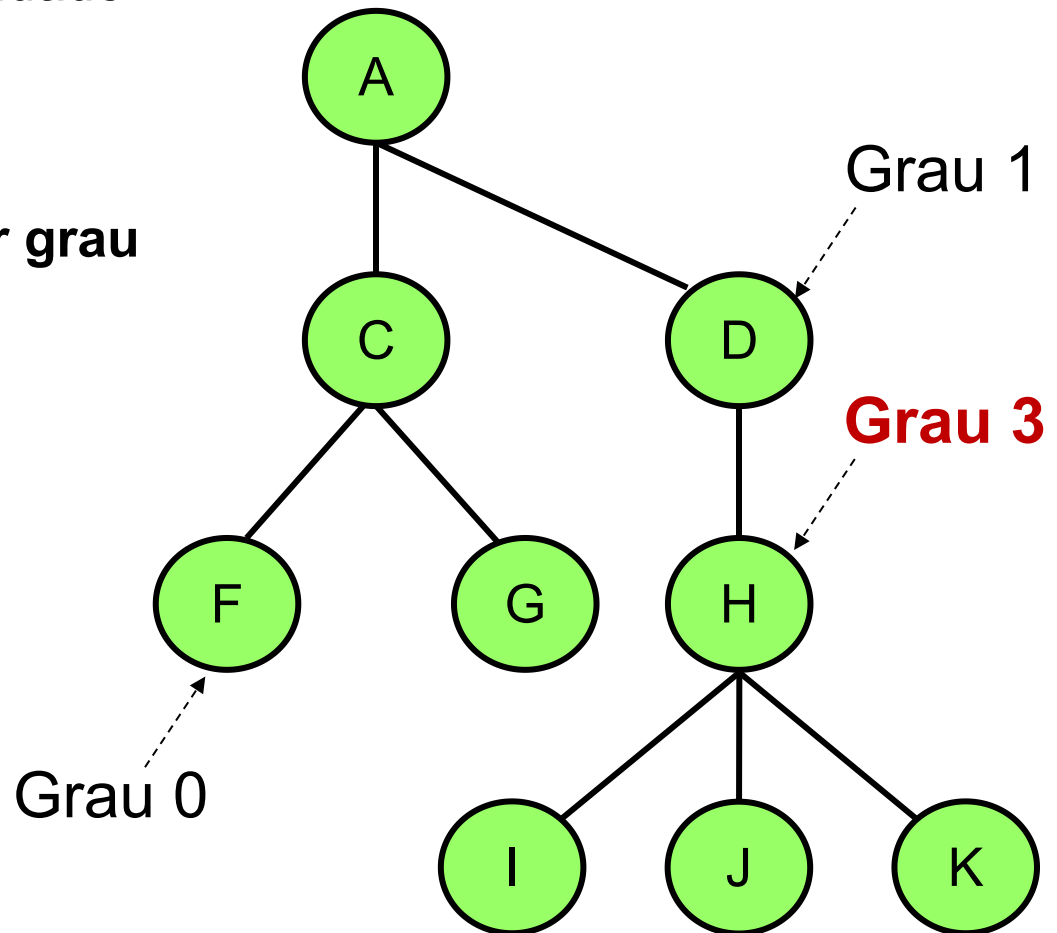


# Árvores: exemplo

O grau de um nó é a quantidade de filhos do nó

O grau da árvore é o maior grau entre seus nós

**Grau da árvore = 3**



# Tipos de árvores

---

**Floresta**

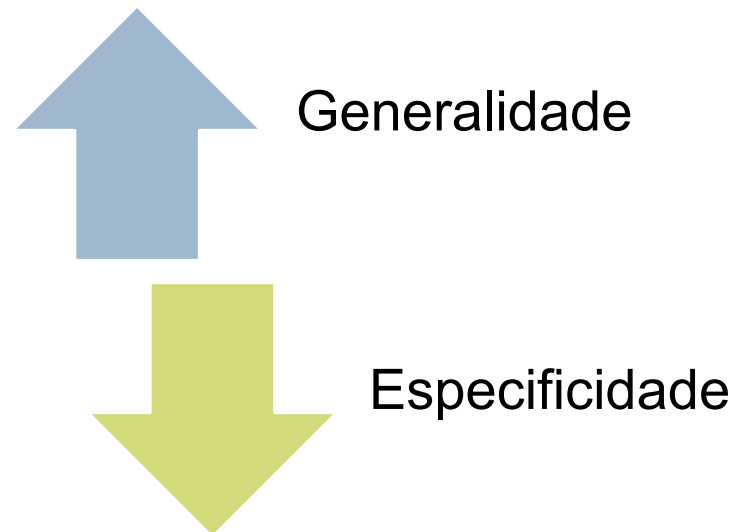
**Árvore livre**

**Árvore com raiz**

**Árvore ordenada**

**Árvore n-ária**

**Árvore binária**

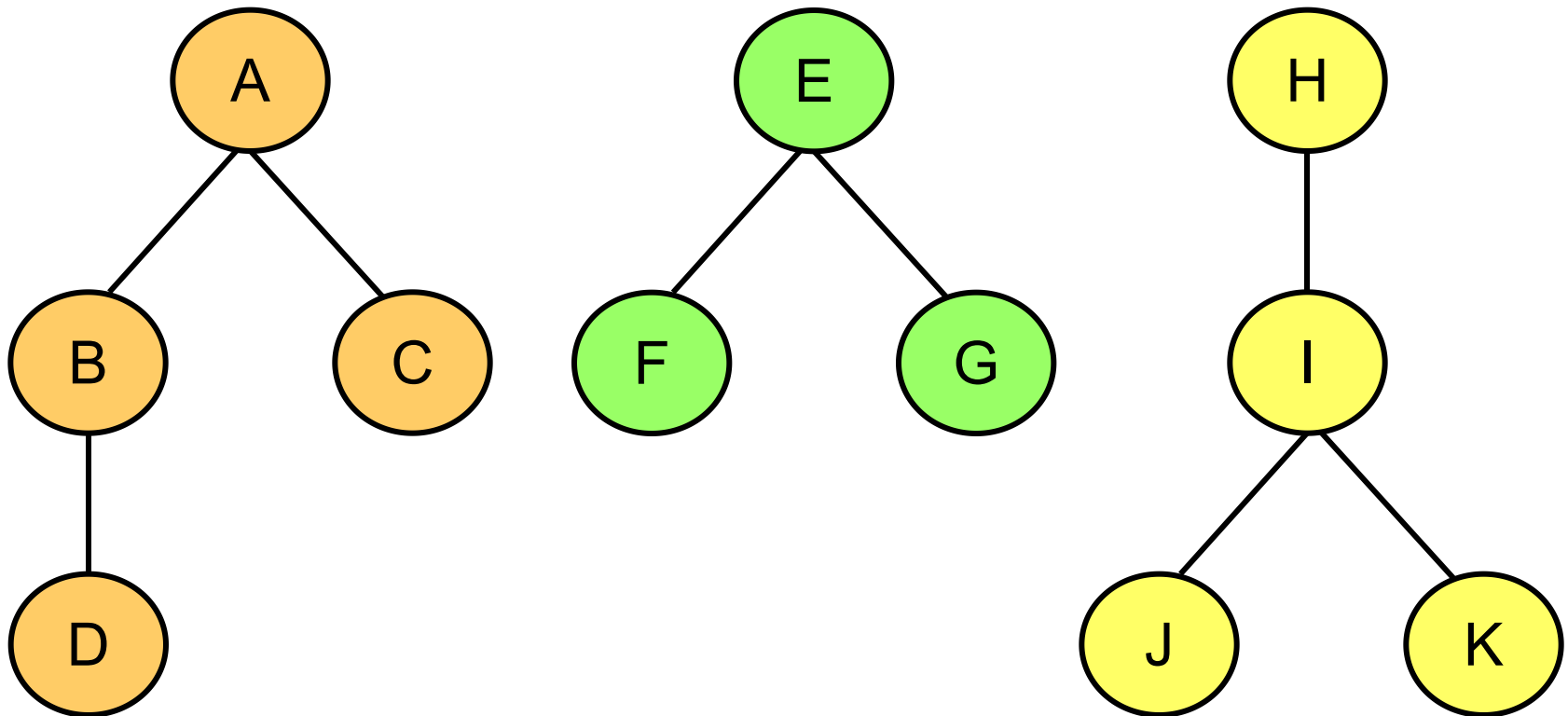




# Tipos de árvore: floresta

**Floresta** é um **grafo acíclico** e **desconexo**

**Conjunto de zero ou mais árvores disjuntas**



# Tipos de árvore: árvore livre

---

**Árvore livre** é um **grafo  $G$  acíclico, conectado e não direcionado** que satisfaz alguns requisitos:

Um par de vértices qualquer está **ligado por apenas um caminho**

**$G$  é conexo** e a remoção de uma aresta desconecta  **$G$**

**$G$  é acíclico** e a inclusão de uma nova aresta cria um ciclo em  **$G$**

Árvore com  **$N$**  vértices possui  **$(N - 1)$**  arestas

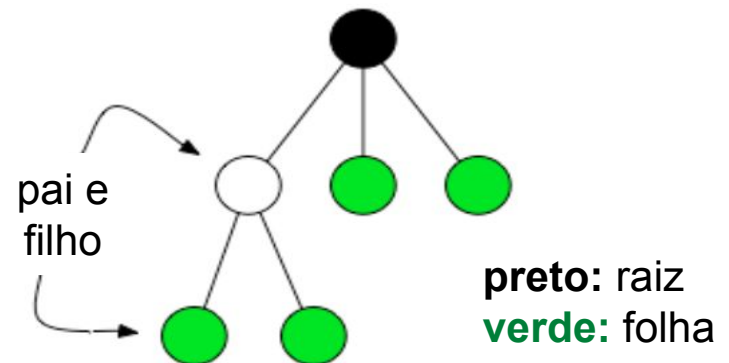
# Tipos de árvore: árvore com raiz

Esse tipo de árvore possui um vértice diferenciado em relação aos demais (**nó raiz**):

Qualquer nó é raiz de sua subárvore  
Formada pelos seus descendentes

Qualquer nó está conectado a um vértice superior (**nó pai**)  
**Exceto o nó raiz**

Possibilita relação hierárquica



# Tipos de árvore: árvore ordenada

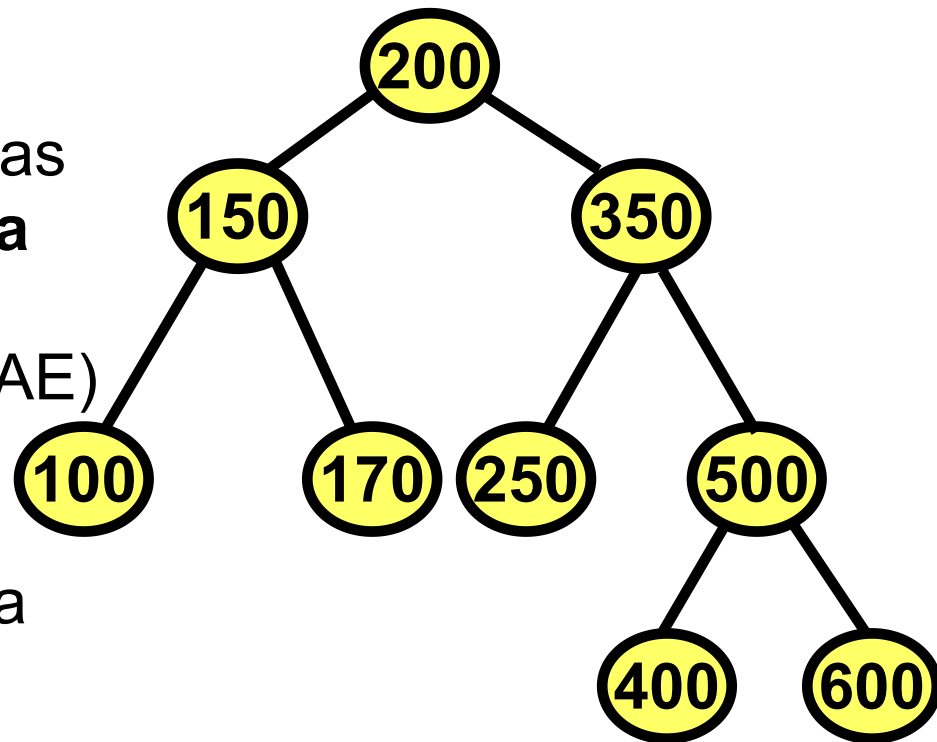
**Árvore ordenada** é uma árvore com raiz cuja disposição dos nós é significativa

Ordem das suas subárvores afeta o funcionamento da aplicação

Normalmente são ordenadas **da esquerda para a direita**

Subárvore da esquerda (SAE) **diferente** da direita (SAD)

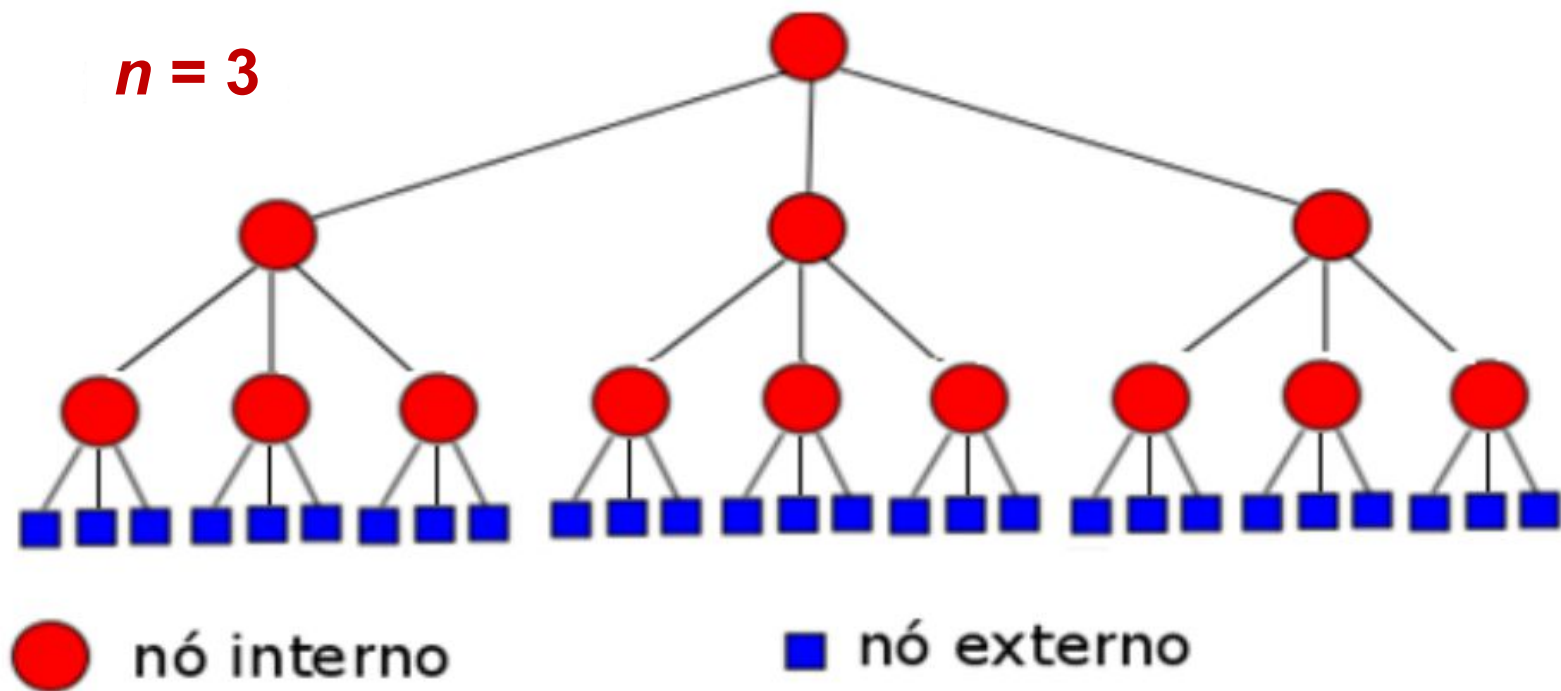
**Ex:** árvore de busca binária



# Tipos de árvore: árvore n-ária

**Árvore n-ária** é uma árvore com raiz onde cada nó possui até  $n$  filhos

$n$  é o **grau máximo dos nós da árvore**



# Tipos de árvore: árvore binária

---

**Árvore binária** é um caso particular de árvore n-ária onde cada nó tem no máximo 2 filhos ( **$n = 2$** )

# Tipos de árvore: árvore binária

---

**Árvore binária** é um caso particular de árvore n-ária onde cada nó tem no máximo 2 filhos ( **$n = 2$** )

Tipo de árvore bastante explorada na computação  
**Facilidade de implementação**

# Tipos de árvore: árvore binária

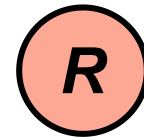
---

**Árvore binária** é um caso particular de árvore n-ária onde cada nó tem no máximo 2 filhos ( **$n = 2$** )

Tipo de árvore bastante explorada na computação  
**Facilidade de implementação**

**Representação:**

Nó raiz ( **$R$** )





# Tipos de árvore: árvore binária

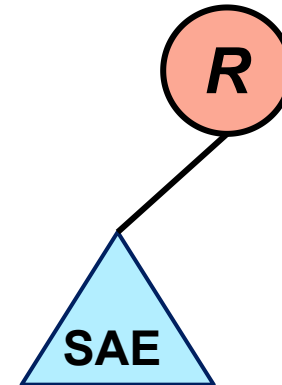
**Árvore binária** é um caso particular de árvore n-ária onde cada nó tem no máximo 2 filhos ( **$n = 2$** )

Tipo de árvore bastante explorada na computação  
**Facilidade de implementação**

## Representação:

Nó raiz ( **$R$** )

Subárvore da esquerda ( **$SAE$** )



# Tipos de árvore: árvore binária

**Árvore binária** é um caso particular de árvore n-ária onde cada nó tem no máximo 2 filhos ( $n = 2$ )

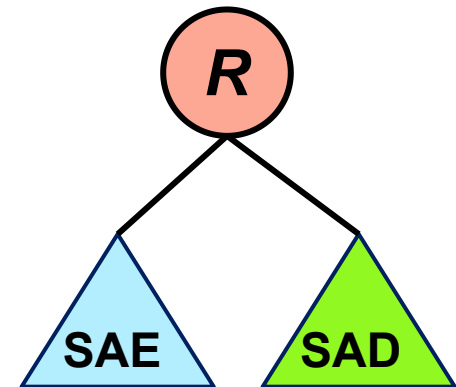
Tipo de árvore bastante explorada na computação  
**Facilidade de implementação**

## Representação:

Nó raiz (**R**)

Subárvore da esquerda (**SAE**)

Subárvore da direita (**SAD**)



# Tipos de árvore: árvore binária

**Árvore binária** é um caso particular de árvore n-ária onde cada nó tem no máximo 2 filhos ( $n = 2$ )

Tipo de árvore bastante explorada na computação  
**Facilidade de implementação**

## Representação:

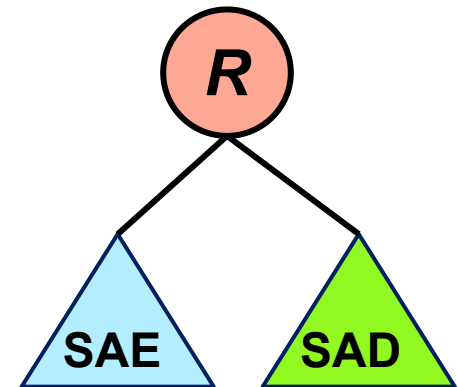
Nó raiz (**R**)

Subárvore da esquerda (**SAE**)

Subárvore da direita (**SAD**)

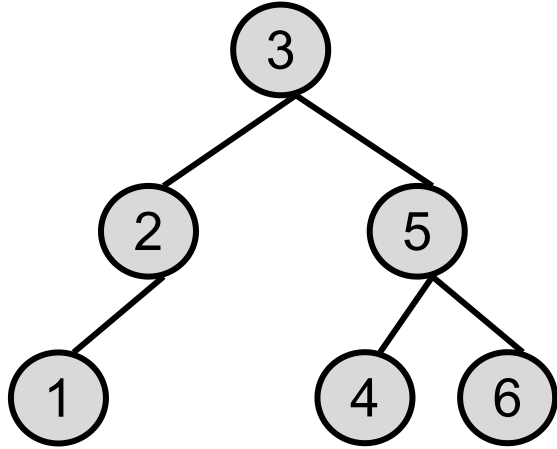
**Notação textual:** < **R** **SAE** **SAD** >

Árvore vazia: < >



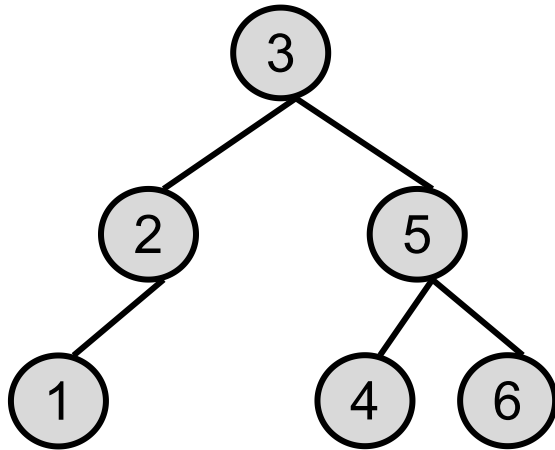
# Exemplo árvore binária

---



# Exemplo árvore binária

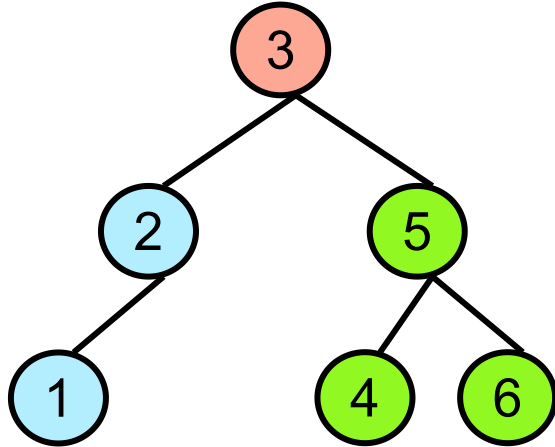
---



**Notação textual:**

<3<2<1<><>><>><5<4<><>><6<><>>>>

# Exemplo árvore binária

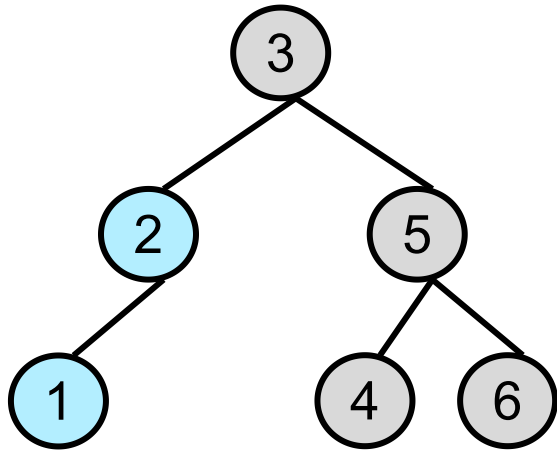


## Notação textual:

<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>

# Exemplo árvore binária



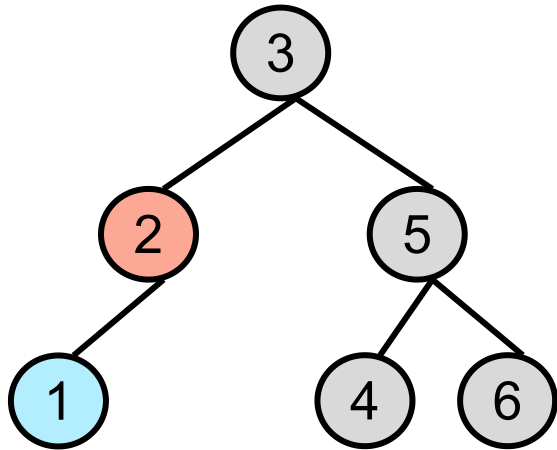
## Notação textual:

<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>

# Exemplo árvore binária



## Notação textual:

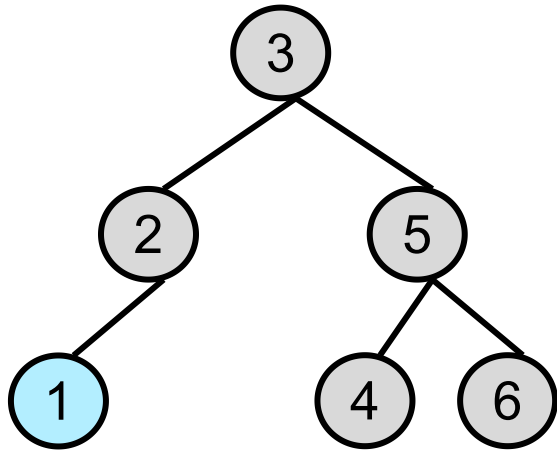
<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>



# Exemplo árvore binária



## Notação textual:

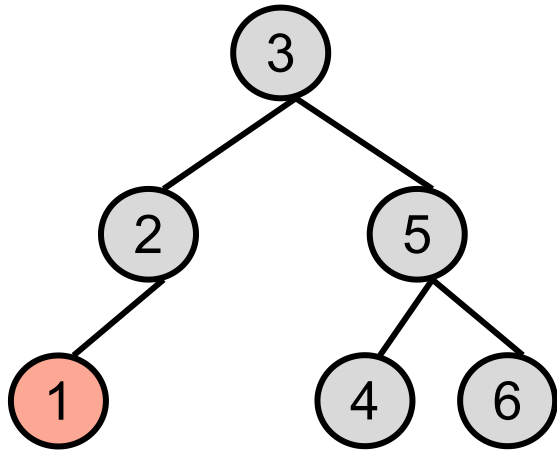
<3<2<1<><>><>><5<4<><>><6<><>>>>

<**3**<**2**<**1**<><>><>><**5**<**4**<><>><**6**<><>>>>

<3<**2**<**1**<><>><>><5<4<><>><6<><>>>>

<3<2<**1**<><>><>><5<4<><>><6<><>>>>

# Exemplo árvore binária



## Notação textual:

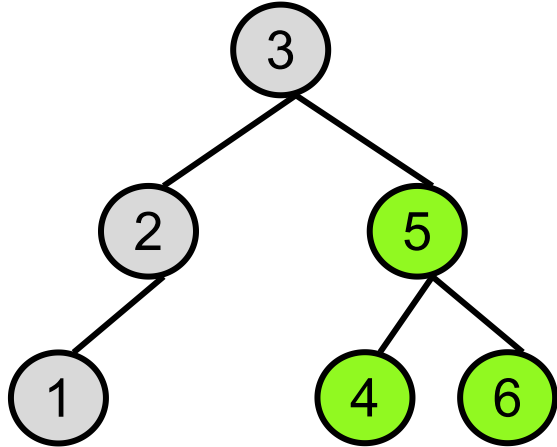
<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>

# Exemplo árvore binária



## Notação textual:

<3<2<1<><>><>><5<4<><>><6<><>>>>

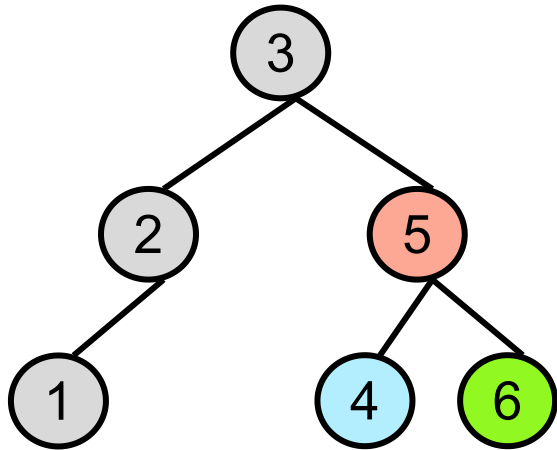
<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>

# Exemplo árvore binária



## Notação textual:

<3<2<1<><>><>><5<4<><>><6<><>>>>

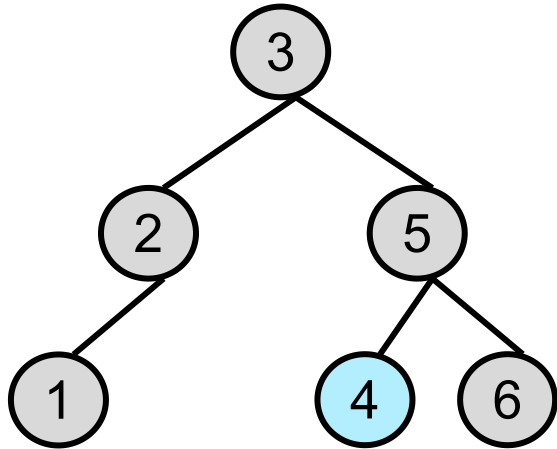
<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>

# Exemplo árvore binária



## Notação textual:

<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>

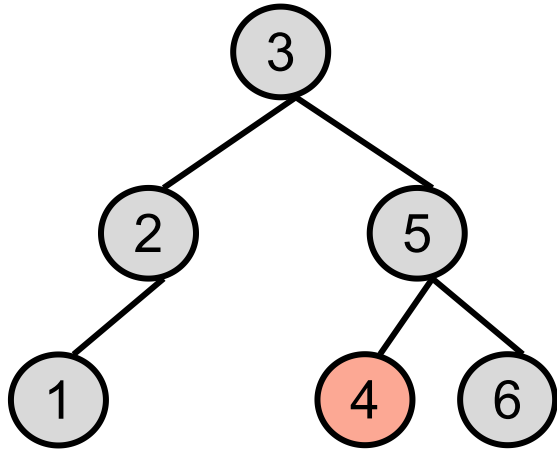
<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>

# Exemplo árvore binária



## Notação textual:

<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>

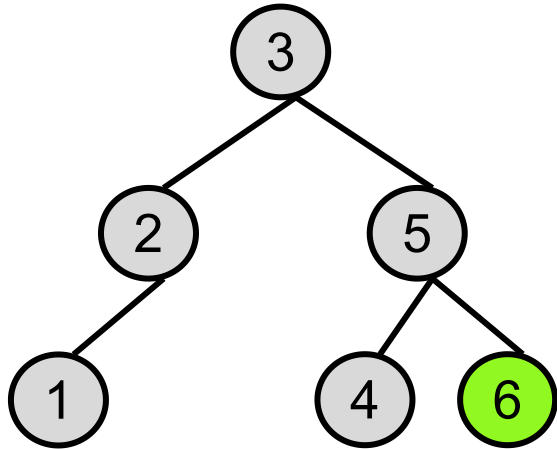
<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>

# Exemplo árvore binária



## Notação textual:

<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>

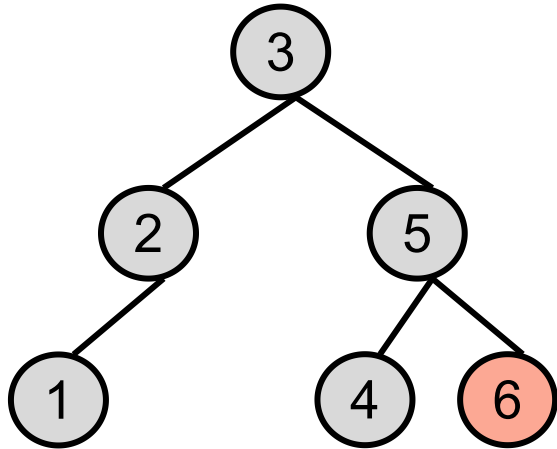
<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>

# Exemplo árvore binária



## Notação textual:

<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>

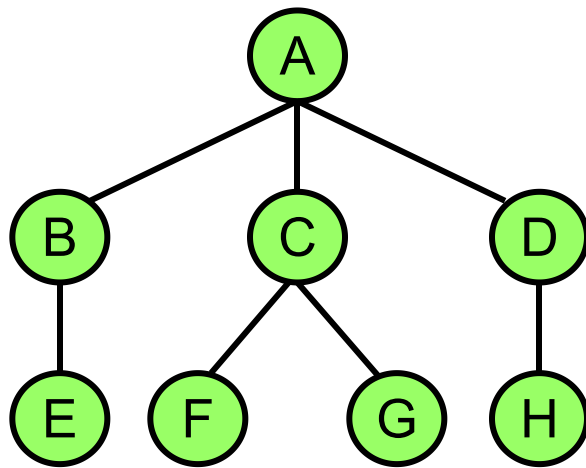
<3<2<1<><>><>><5<4<><>><6<><>>>>

<3<2<1<><>><>><5<4<><>><6<><>>>>

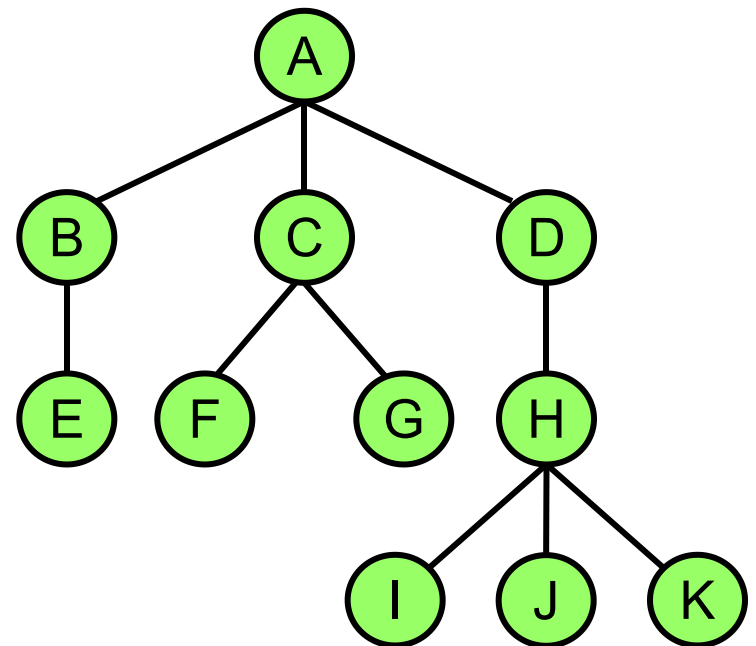


# Outras definições

Uma árvore é **balanceada** quando **todos os nós folha** (terminais) **estão à mesma distância do nó raiz**



**Balanceada**



**Não balanceada**

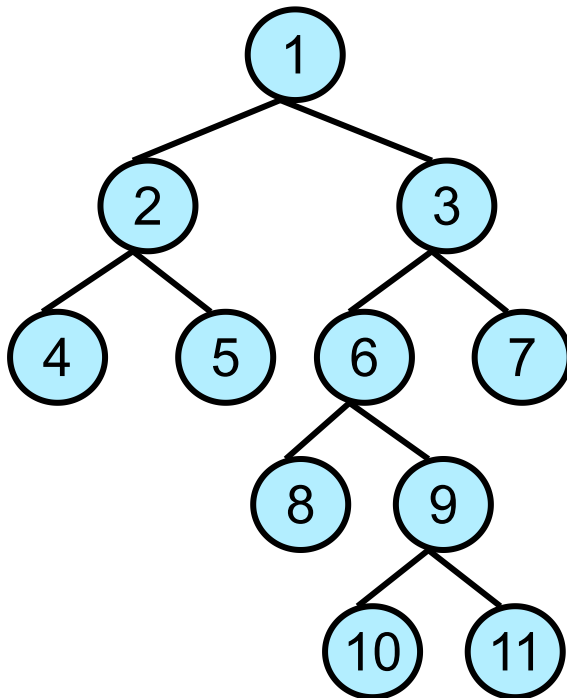
# Outras definições

**Árvore estritamente binária** é uma árvore em que:

Cada nó tem **0 ou 2 filhos** (Nº máximo de filhos)

Nós com ZERO filhos são terminais

Nós com 2 filhos são nós internos



**Não existe controle dos níveis** que estão os nós terminais

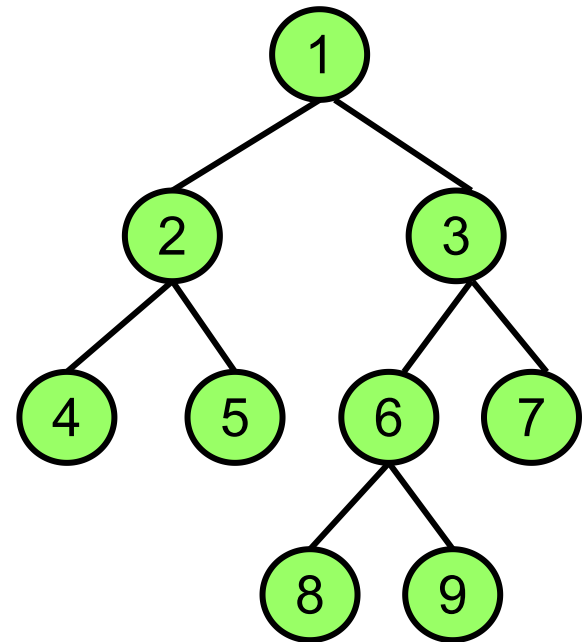
# Outras definições

**Árvore completa** é uma árvore em que:

Nós intermediários têm o número máximo de filhos

Nós terminais ocupam apenas o penúltimo ou o último nível da árvore

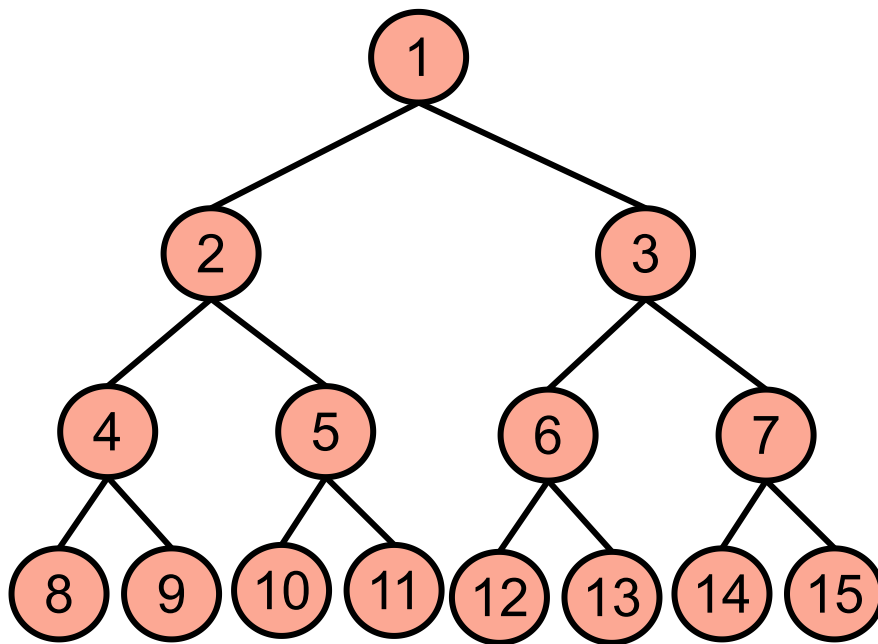
O **Nº de nós terminais** de uma árvore binária completa é o **Nº de nós intermediários + 1**



# Outras definições

**Árvore cheia** é uma **árvore completa e balanceada**

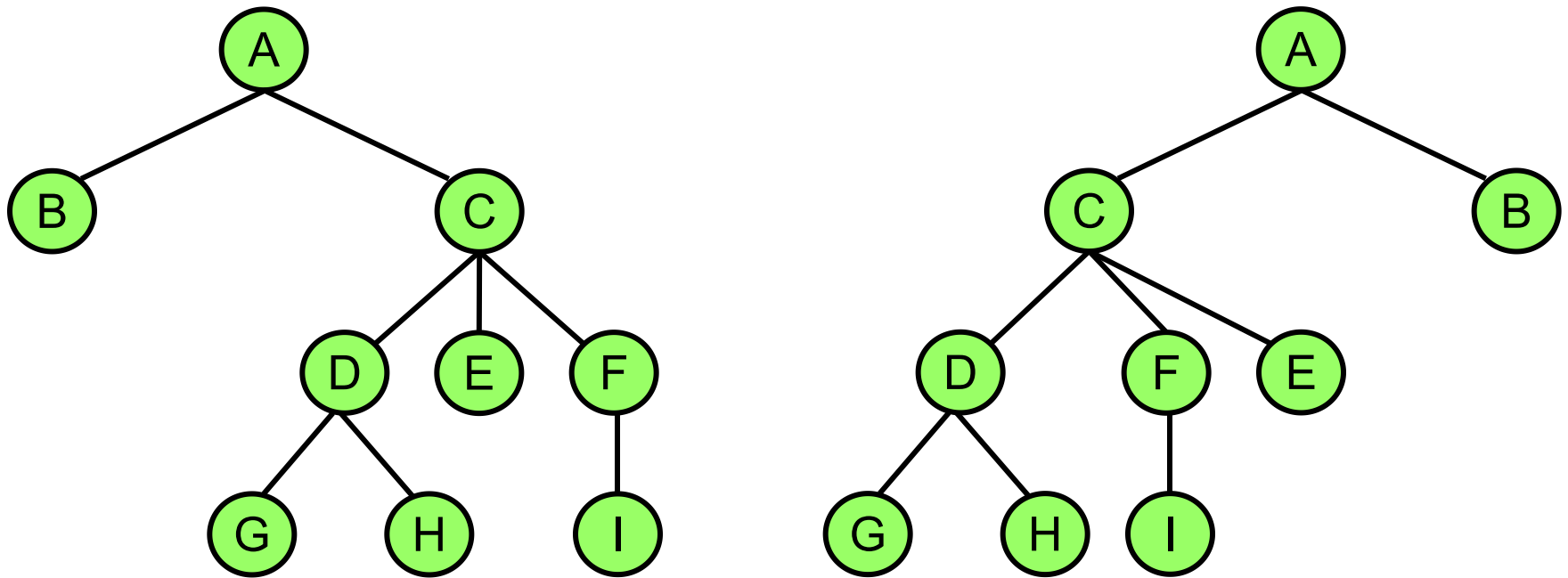
Todos os nós terminais têm a **mesma profundidade**



O **Nº de nós internos** de uma **árvore binária cheia** é  **$2^h - 1$**  (sendo  **$h$**  a altura da árvore)

# Outras definições

Duas árvores são **isomorfas** se podem se tornar coincidentes através da permutação na ordem das subárvores de seus nós



# Exercícios

---

1. Qual é o menor número de níveis que uma árvore binária com 42 nós pode ter? E o maior?
2. Qual o número máximo de nós no nível 5 de uma árvore com aridade 3 (árvore  $n$ -ária com  $n = 3$ )?
3. Apresente a representação hierárquica e do diagrama de inclusão (Venn) da árvore ( 1 ( 2 5 ( 3 ( 7 8 ) 9 ) 6 ( 4 10 ( 11 ) ) 12 ) ).
4. Apresente o comprimento dos caminhos entre o nó raiz e os nós folhas da árvore do exercício anterior. Depois determine a altura e o nível de cada nó da árvore, e a altura e a profundidade da árvore.

# Bibliografia

---

Slides adaptados do material do Prof. Dr. Bruno Travençolo, da Profa. Dra. Gina Maira Barbosa de Oliveira e da Profa. Dra. Denise Guliato.

EDELWEISS, N; GALANTE, R. Estruturas de dados (Série Livros Didáticos Informática UFRGS, v. 18), Bookman, 2008.

CORMEN, T.H. et al. Algoritmos: Teoria e Prática, Campus, 2002

ZIVIANI, N. Projeto de algoritmos: com implementações em Pascal e C (2ª ed.), Thomson, 2004

MORAES, C.R. Estruturas de Dados e Algoritmos: uma abordagem didática (2ª ed.), Futura, 2003