

Revisão de Listas Lineares, Pilhas, Filas e Busca Binária e Introdução à Árvores

Sumário

- Referência Bibliografia da Aula
- Repositório *Git*
- Recapitulando...
- Listas Lineares em Alocação Sequencial
- Ponteiros e Alocação de Memória
- Recapitulando...
- Listas Lineares em Alocação Encadeada



Sumário

- Listas simplesmente Encadeadas
- Listas Circulares Encadeadas
- Listas Duplamente Encadeadas
- Recapitulando...
- Busca Binária em Lista Lineares Encadeadas
- Árvores
- Representação de uma Árvore

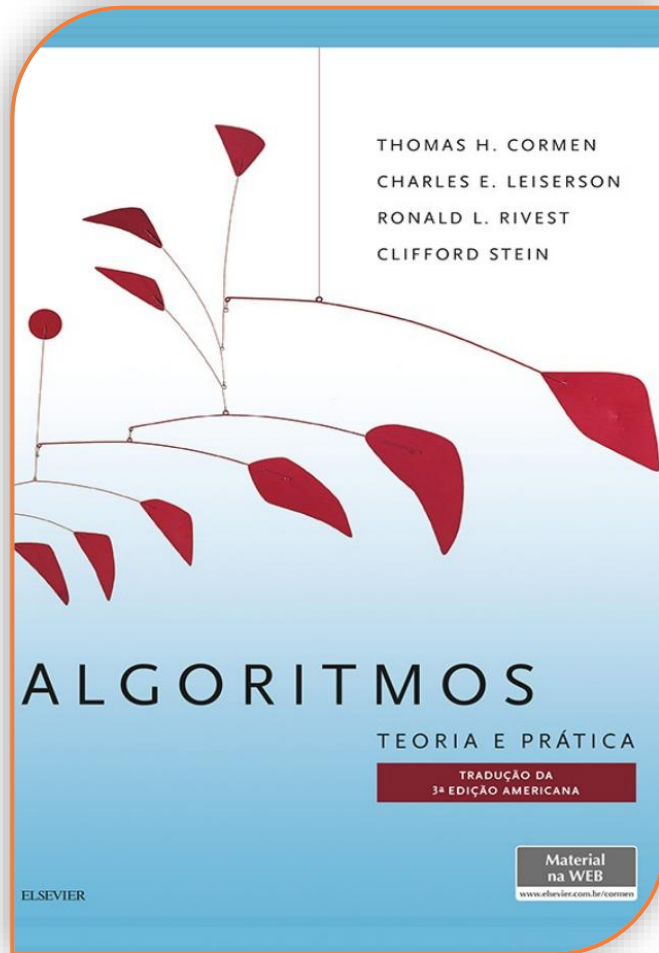



Sumário

- Elementos de uma Árvore
- Árvores Binárias
- Percurso em Árvores Binárias



Referência Bibliografia da Aula

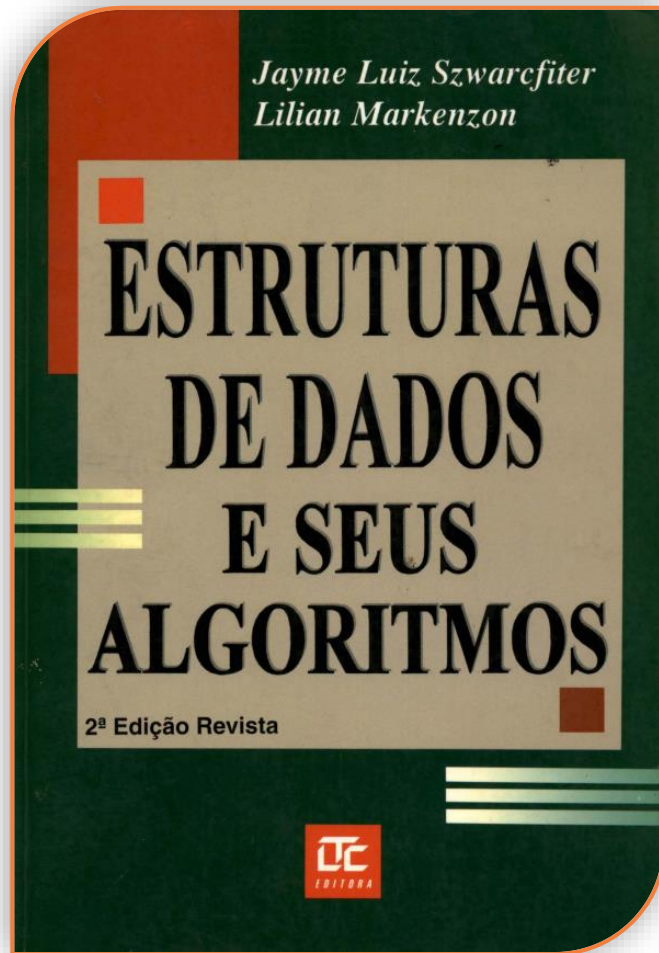


-  CORMEN, T.H. et al. **Algoritmos: Teoria e Prática**. 3ª ed. Rio de Janeiro: Elsevier, 2012.



ESTRUTURA DE DADOS E PARADIGMAS
REVISÃO DE LISTAS LINEARES, PILHAS, FILAS E BUSCA BINÁRIA E INTRODUÇÃO À ÁRVORES

Referência Bibliografia da Aula




-  SZWACTFITER, J.L. et al. **Estruturas de Dados e Seus Algoritmos**. 2ª ed. Editora LTC, 1997.




ESTRUTURA DE DADOS E PARADIGMAS
REVISÃO DE LISTAS LINEARES, PILHAS, FILAS E BUSCA BINÁRIA E INTRODUÇÃO À
ÁRVORES

Objetivo Geral da Aula

-  Os estudantes devem conhecer a **Estrutura de Dados** do tipo **Lista**, compreendendo seu funcionamento interno e sabendo aplica-la na resolução de problemas.




Objetivo Geral da Aula

-  Os estudantes devem conhecer as **Estruturas de Dados** dos tipos **Pilhas** e **Filas**, suas diferentes formas de implementação sabendo aplica-las na resolução de problemas.




Objetivo Geral da Aula

-  Os estudantes devem conhecer a técnica de **Busca Binária**, compreendendo o quanto ela pode acelerar as buscas, suas limitações, formas de implementação, e saber aplicá-la na resolução de problemas reais.




Objetivo Geral da Aula

-  Os estudantes devem conhecer o conceito de **Árvores** como uma **Estrutura de Dados hierárquica**, seu uso para estruturar informações para acelerar buscas, inserção de elementos, e saber aplicá-la na resolução de problemas.



Repositório *Git*

- github.com/whoisraibolt/UNIFESO-CCOMP-EDP
-  Repositório com o conteúdo da disciplina **Estrutura de Dados e Paradigmas** do curso **Ciência da Computação** da **UNIFESO** — Centro Universitário Serra dos Órgãos.



ESTRUTURA DE DADOS E PARADIGMAS
REVISÃO DE LISTAS LINEARES, PILHAS, FILAS E BUSCA BINÁRIA E INTRO
ÁRVORES



Recaptulando...

- **Estruturas de Dados:**

- Empregada na representação do **modelo matemático**.
- **Formas de organizar dados** de modo a atender aos **diferentes requisitos de processamento**, levando em consideração suas **relações lógicas**.



Recaptulando...



ESTRUTURA DE DADOS E PARADIGMAS
REVISÃO DE LISTAS LINEARES, PILHAS, FILAS E BUSCA BINÁRIA E INTRODUÇÃO À
ÁRVORES



Recaptulando...



ESTRUTURA DE DADOS E PARADIGMAS
REVISÃO DE LISTAS LINEARES, PILHAS, FILAS E BUSCA BINÁRIA E INTRODUÇÃO À
ÁRVORES



Recaptulando...

- Uma **Lista Linear** é uma Estrutura de Dados Linear de manipulação mais simples, que agrupa informações referentes a um conjunto de elementos que, de alguma forma, se relacionam entre si.
- **Exemplos:**
 - **Notas de alunos, informações sobre clientes de uma empresa, entre outros.**

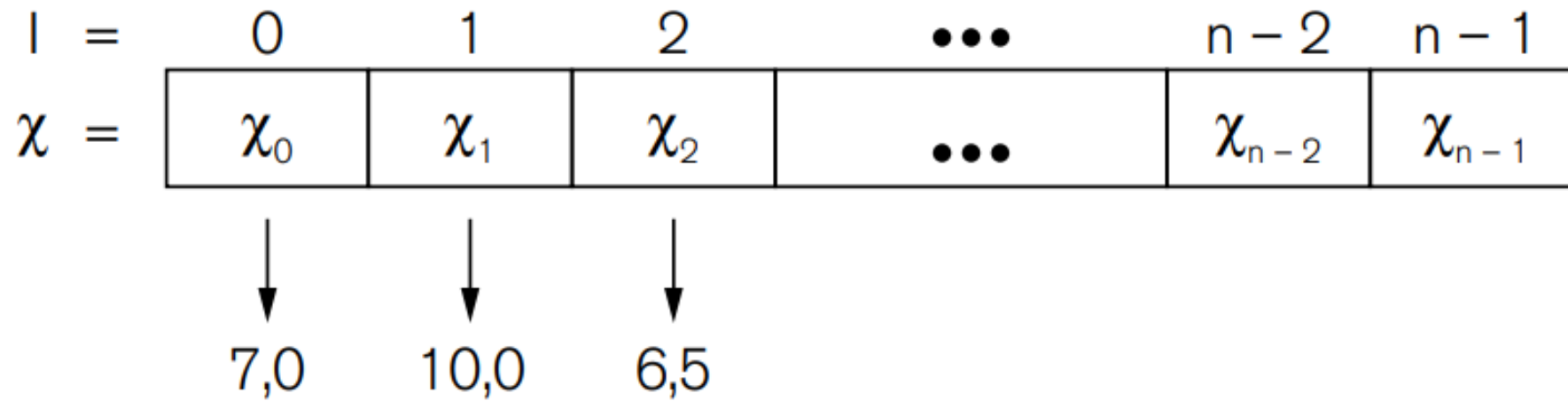


Recaptulando...

- Na **Lista Linear** cada **nó** é **precedido** por um **nó** e **sucedido** por outro **nó**.
- **Exceto** o **primeiro nó**, que não possui **predecessor** e o **último nó**, que não possui **sucessor**.



Recaptulando...



Cliente				
Cód. Cliente	Nome	Endereço	Fone	RG



Recaptulando...

- Em uma **Lista Linear**, cada **nó** é formado por **campos**.
- Que armazenam as características distintas dos elementos da lista.

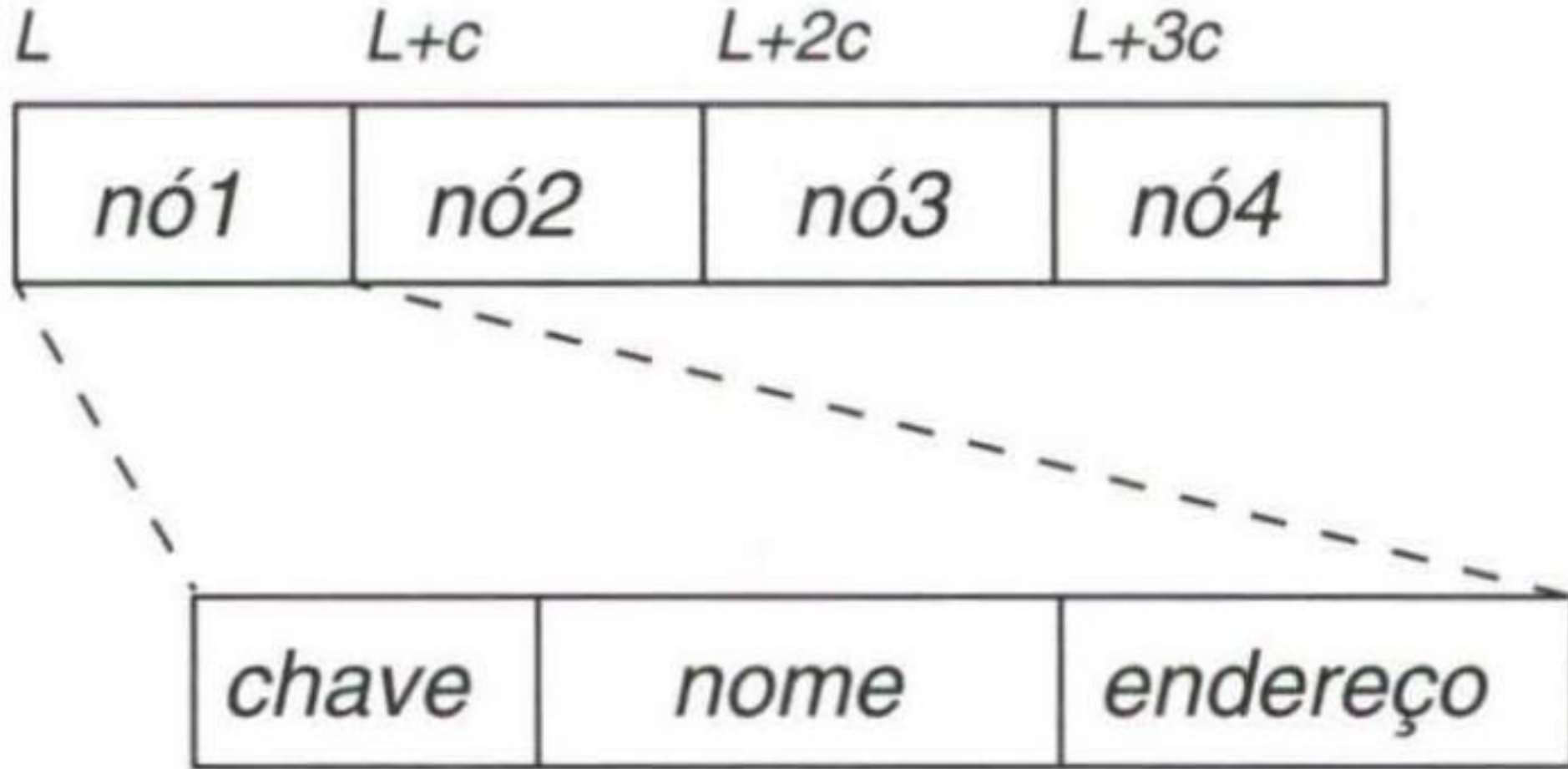


Recaptulando...

- Cada **nó** possui:
 - **Nome**: Nome da Lista.
 - **Chave**: **Identificador**; **elemento**; **valor** a ser encontrado.
 - **Endereço**: Posição; índice em que a **Chave** se encontra.



Recaptulando...



Recaptulando...

- **Tipos de Armazenamento:**

- Podem ser classificados de acordo com a **posição relativa** na memória, ou seja, posição contígua ou posição não contígua de **dois nós consecutivos** na lista.
 - **Alocação Sequencial** (posição contígua);
 - **Alocação Encadeada** (posição não contígua).



Listas Lineares em Alocação Sequencial

- Quando temos uma **Lista** em que todos os elementos são armazenados em um espaço de **memória contíguo**, dizemos que a **Lista** tem **alocação sequencial**.
- Na Alocação Sequencial considera-se sempre, a **primeira posição** da lista no **endereço 1** da memória disponível.
- Endereço **0** em **Python**.



Listas Lineares em Alocação Sequencial



```
# Lista Linear em Alocação Sequencial  
lista = [11, 4, 9, 15, 16, 18, 18, 13, 2, 10]  
  
# Acesso Aleatório  
print(lista[7])
```



Ponteiros e Alocação de Memória

- Existem estruturas que fazem uso de **indicadores especiais**, denominados **ponteiros**, para o acesso a **posições selecionadas**.



Recaptulando...



ESTRUTURA DE DADOS E PARADIGMAS
REVISÃO DE LISTAS LINEARES, PILHAS, FILAS E BUSCA BINÁRIA E INTRODUÇÃO À
ÁRVORES

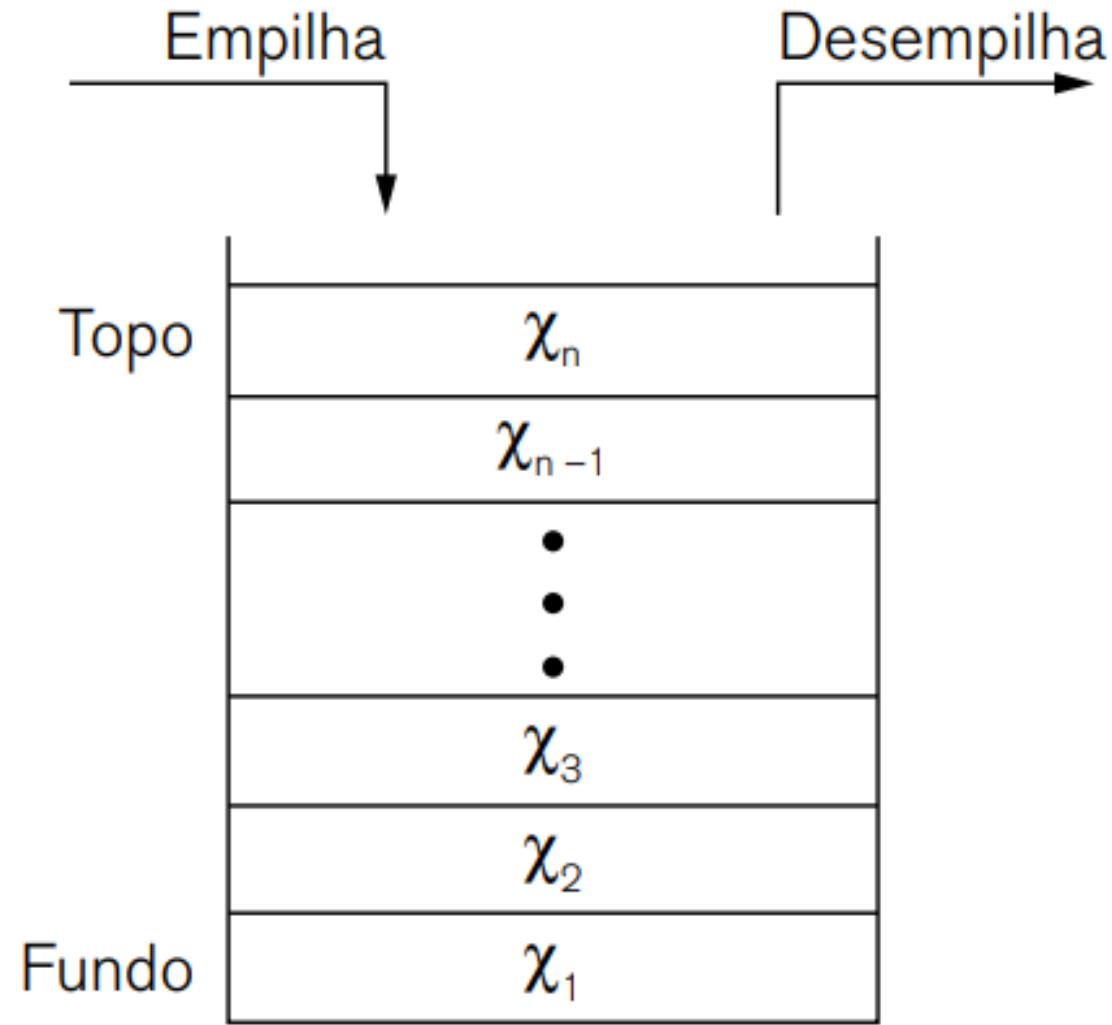


Recaptulando...

- No caso da **Pilha**, apenas um ponteiro precisa ser considerado:
 - **Ponteiro topo.**
- As **inserções e remoções** são **executadas** na **mesma extremidade** da lista.



Recaptulando...



Recaptulando...

- Como o **último elemento** que entrou na Pilha será o **primeiro** a **sair** da Pilha, a Pilha é conhecida como uma estrutura do tipo **LIFO** (**Last In First Out**).
- **Exemplos:**
 - Pilhas de pratos.
 - Chamada de procedimentos: A medida que **procedimentos** chamam outros **procedimentos**, mais e mais **endereços** de retorno devem ser **empilhados**. Estes são **desempilhados** à medida que os procedimentos chegam ao seu **fim**.



Recaptulando...



ESTRUTURA DE DADOS E PARADIGMAS
REVISÃO DE LISTAS LINEARES, PILHAS, FILAS E BUSCA BINÁRIA E INTRODUÇÃO À
ÁRVORES

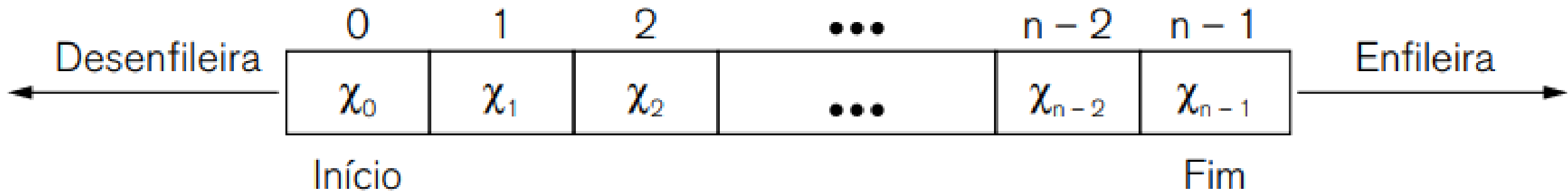


Recaptulando...

- As **Filas** exigem uma implementação um pouco mais elaborada.
- São necessários dois ponteiros:
 - Início de fila (***f***) e;
 - Fim (***r***).
- Para a inserção de um elemento, move-se o ponteiro ***r***.
- Para a remoção, move-se o ponteiro ***f***.



Recaptulando...



Recaptulando...

- As inserções de novos elementos são realizadas numa extremidade da Lista (**Fim**).
- As remoções são feitas na outra extremidade da Lista (Início).
- Uma Fila é uma estrutura do tipo **FIFO** (**First In First Out**).
- Exemplo:
 - Num **sistema operacional**, os **processos prontos** para entrar em execução (aguardando apenas a disponibilidade da **CPU**) são geralmente **mantidos** numa **fila**.



Listas Lineares em Alocação Encadeada

- Dependendo da **quantidade de dados** ou o **uso** de mais de uma **lista**, a **gerência de memória** se torna mais **complexa**.
- Nesses casos se justifica a utilização da **Alocação Encadeada**, também conhecida por **Alocação Dinâmica**.

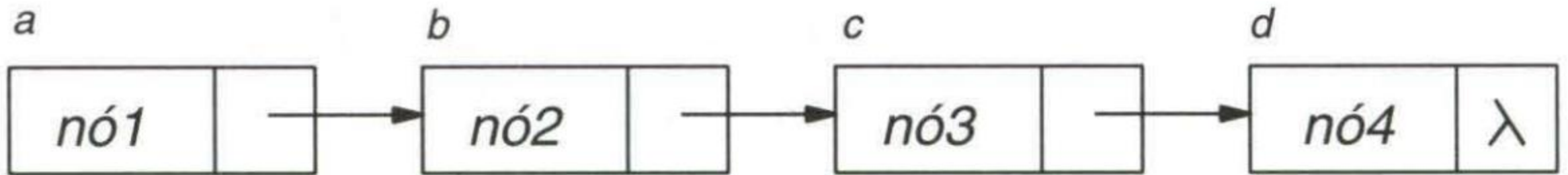
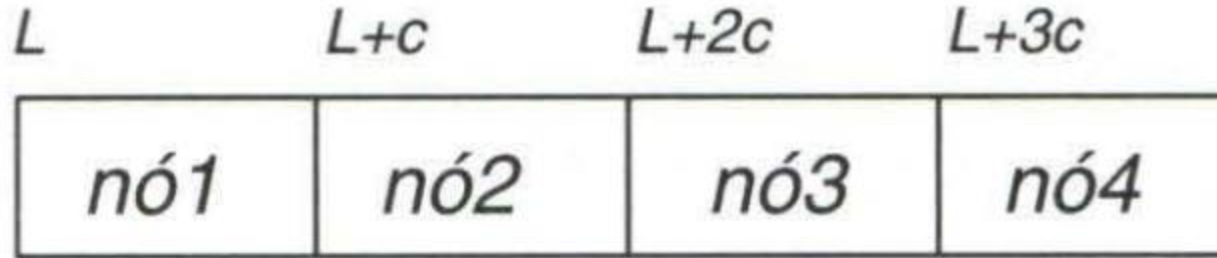


Listas Lineares em Alocação Encadeada

- Uma vez que **posições de memória** são **alocadas** (ou **desalocadas**) na **medida** em que são **necessárias** (ou **dispensadas**).
 - Os **nós** de uma **Lista** encontram-se **aleatoriamente** dispostos na **memória** e são **interligados** por **Ponteiros**.
- É **necessário** o **acréscimo** de um **campo** a cada **nó**, justamente o que indica o **endereço** do **próximo nó** da **Lista**.



Listas Lineares em Alocação Encadeada



Listas simplesmente Encadeadas

- Qualquer **estrutura**, inclusive **Lista**, que seja armazenada em **Alocação Encadeada** requer o uso de um **ponteiro** que indique o endereço de seu primeiro nó (**ptlista**).
- O **percurso** de uma **Lista** é feito então **a partir** desse **ponteiro**.



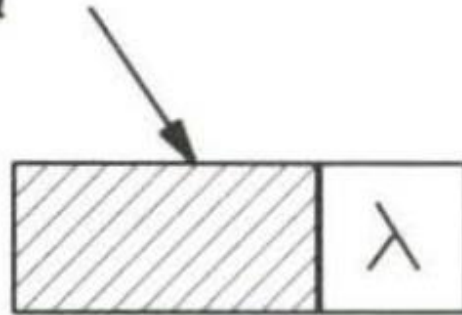
Listas simplesmente Encadeadas

- A ideia consiste em seguir consecutivamente pelos endereços existentes no campo que indica o próximo nó.
- O nó especial **ptlista** não deve conter informações relacionadas à **Lista** propriamente dita.

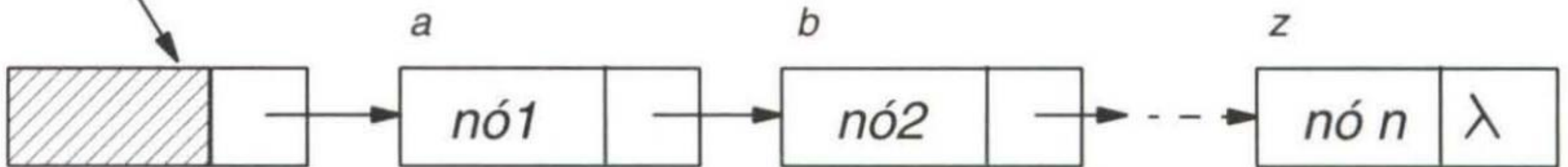


Listas simplesmente Encadeadas

Lista vazia: *ptlista*



ptlista



Listas Circulares Encadeadas

- **Obriga o último nó** da Lista a apontar para o **nó-cabeça (ptlista)**, criando assim uma Lista Circular Encadeada.
- **Exemplos:**
 - **Dias da semana** ou **meses do ano**.

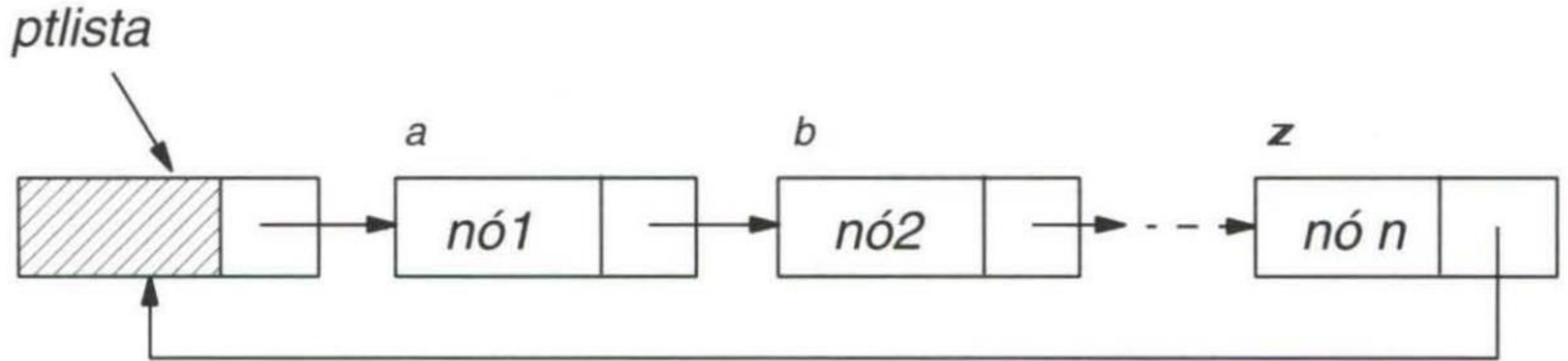


Listas Circulares Encadeadas

- Dessa forma, o **teste de fim de Lista nunca é satisfeito**.
- **Necessário critério de parada** que possa ser incorporado ao teste da busca propriamente dita.



Listas Circulares Encadeadas

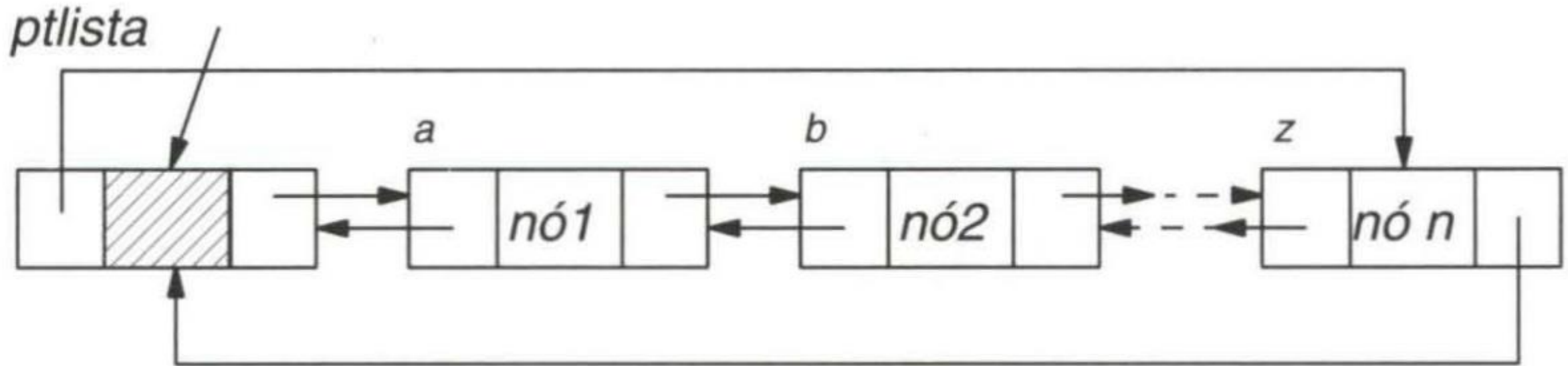


Listas Duplamente Encadeadas

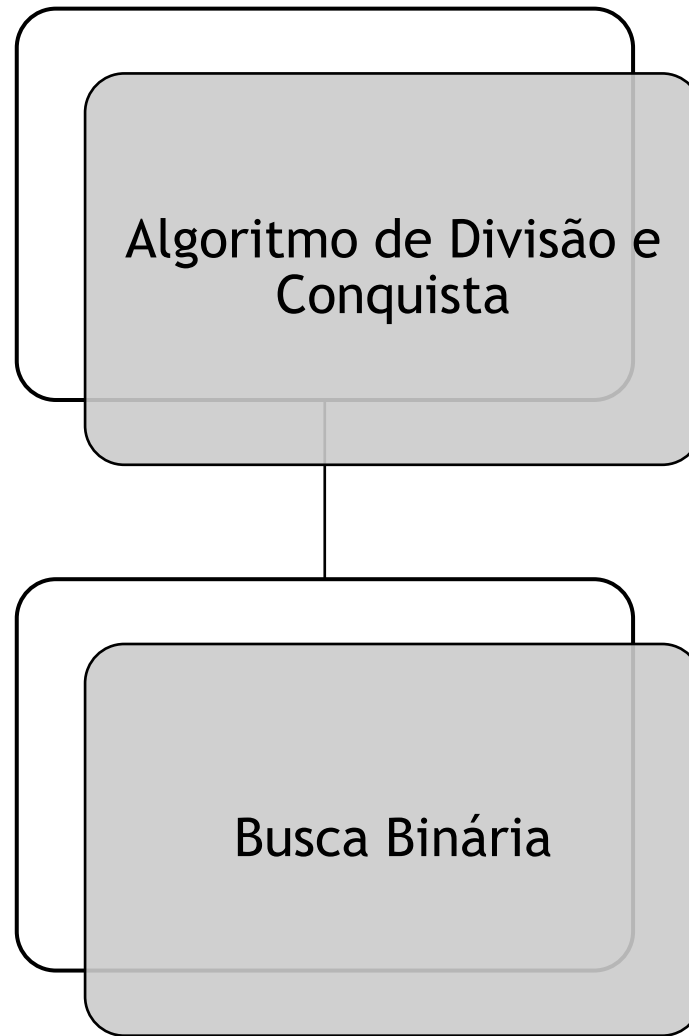
- Cada **nó** tem um **ponteiro** para o **próximo nó** e um **ponteiro** para o **nó anterior**.
- Desta forma, **dado** um **elemento**, podemos **acessar ambos** os elementos **adjacentes**: o **próximo** e o **anterior**.



Listas Duplamente Encadeadas



Recapitulando...



ESTRUTURA DE DADOS E PARADIGMAS
REVISÃO DE LISTAS LINEARES, PILHAS, FILAS E BUSCA BINÁRIA E INTRODUÇÃO À
ÁRVORES



Recapitulando...

- A **Busca Binária** permite percorrer as **Listas**, como se **folheia** uma **lista telefônica**, por exemplo.
- Em uma Lista, o **primeiro nó pesquisado** é o que se encontra no **meio**.
- Se a **comparação não é positiva**, **metade** da **Lista** pode ser **abandonada**, ou seja, o valor procurado se encontra ou na metade inferior (**se for menor**), ou na metade superior (**se for maior**).



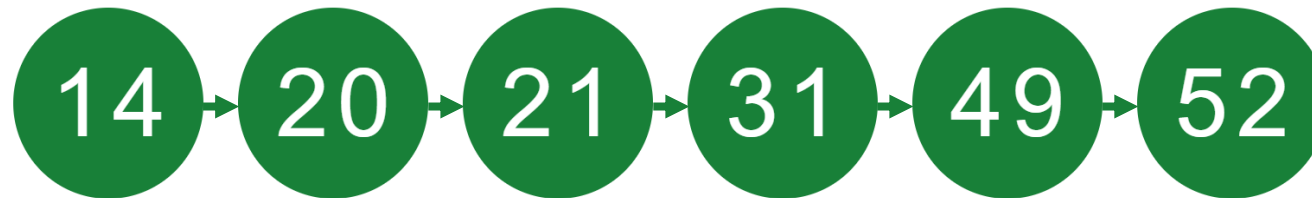
Recapitulando...

- **Busca de um elemento em Listas Lineares em Alocação Sequencial:**
 - A Busca Binária é mais **eficiente**.
- **Entretanto**, apenas com listas lineares em alocação sequencial (estáticas) e **precisamos garantir** que as **chaves estejam ordenadas**.



Busca Binária em Lista Lineares Encadeadas

- Busca de um elemento em Listas Lineares em Alocação Encadeada:
 - Permite que elementos sejam inseridos, substituídos ou excluídos durante a existência da Lista.

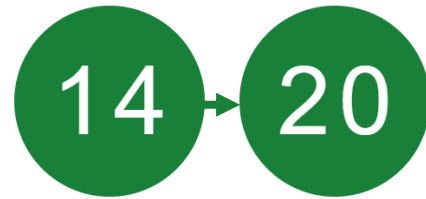


Seria possível?



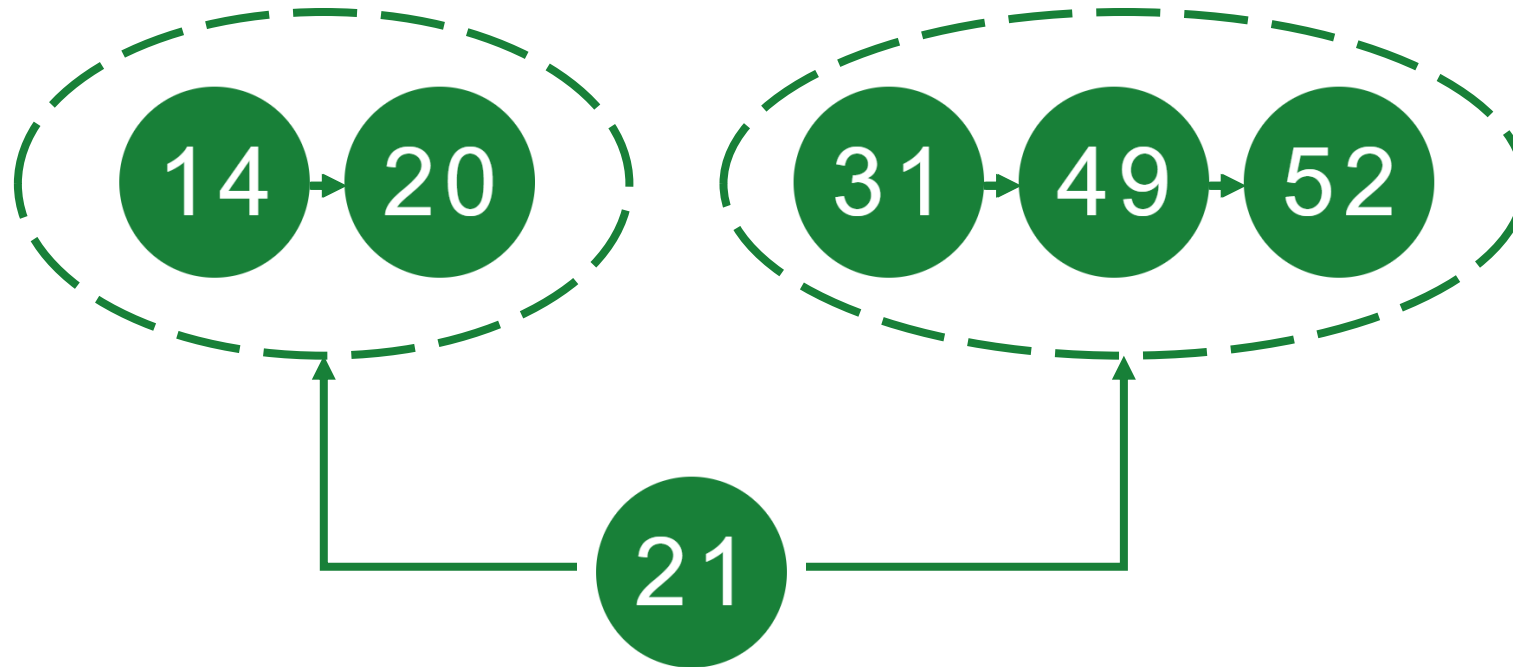
Busca Binária em Lista Lineares Encadeadas

- Busca de um elemento:



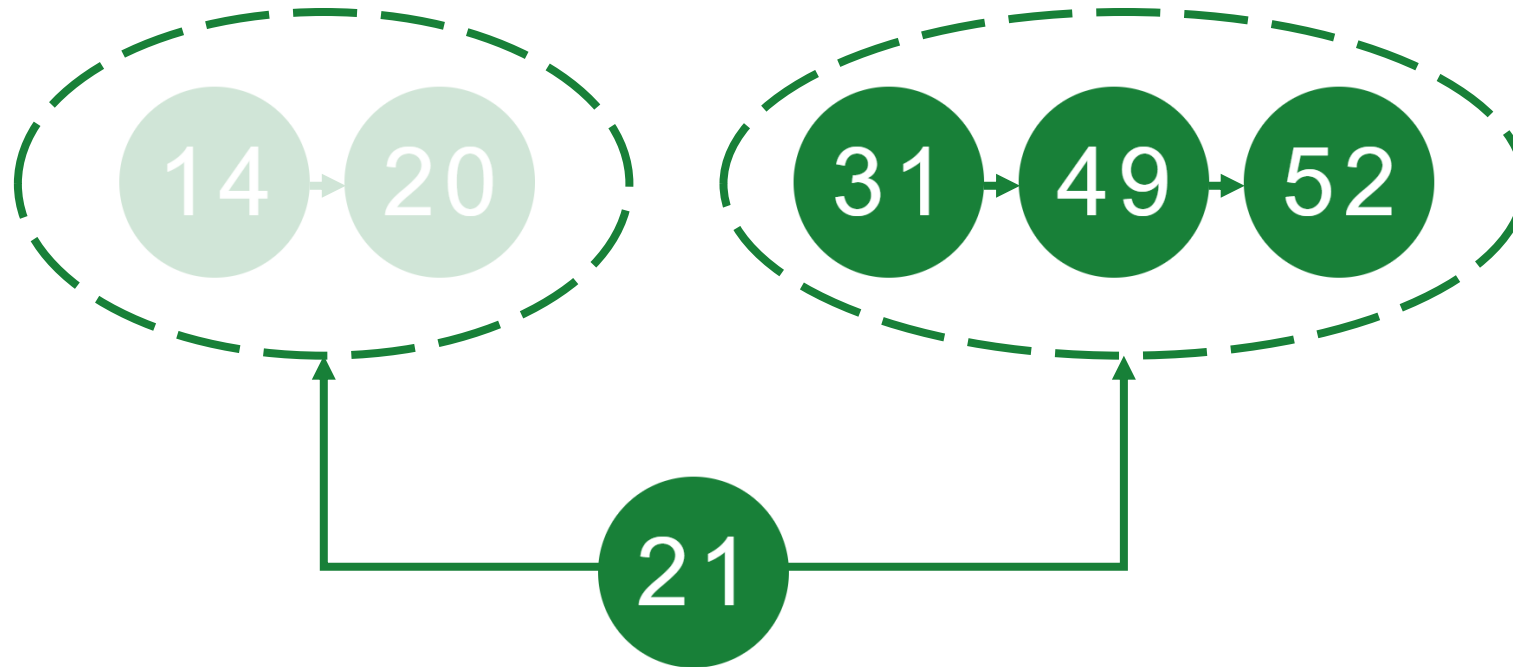
Busca Binária em Lista Lineares Encadeadas

- Busca de um elemento:



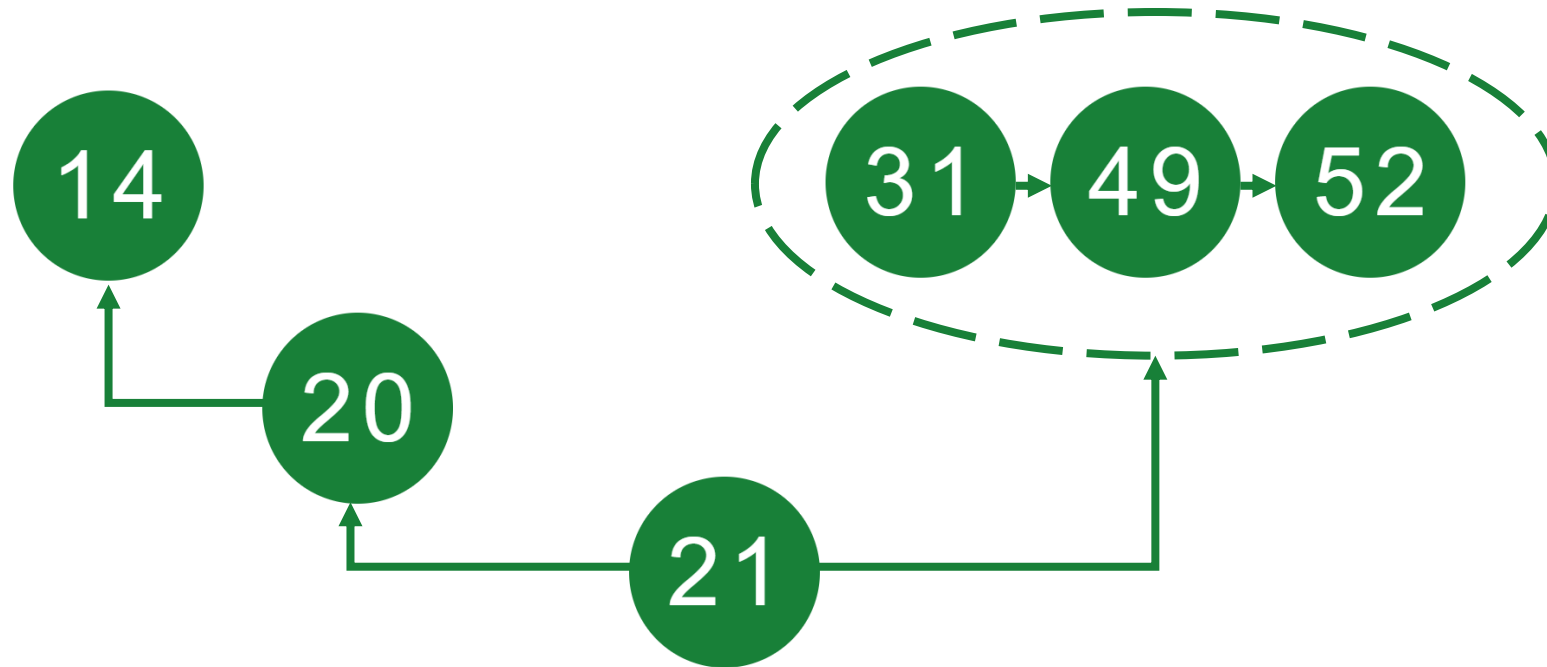
Busca Binária em Lista Lineares Encadeadas

- Busca de um elemento:



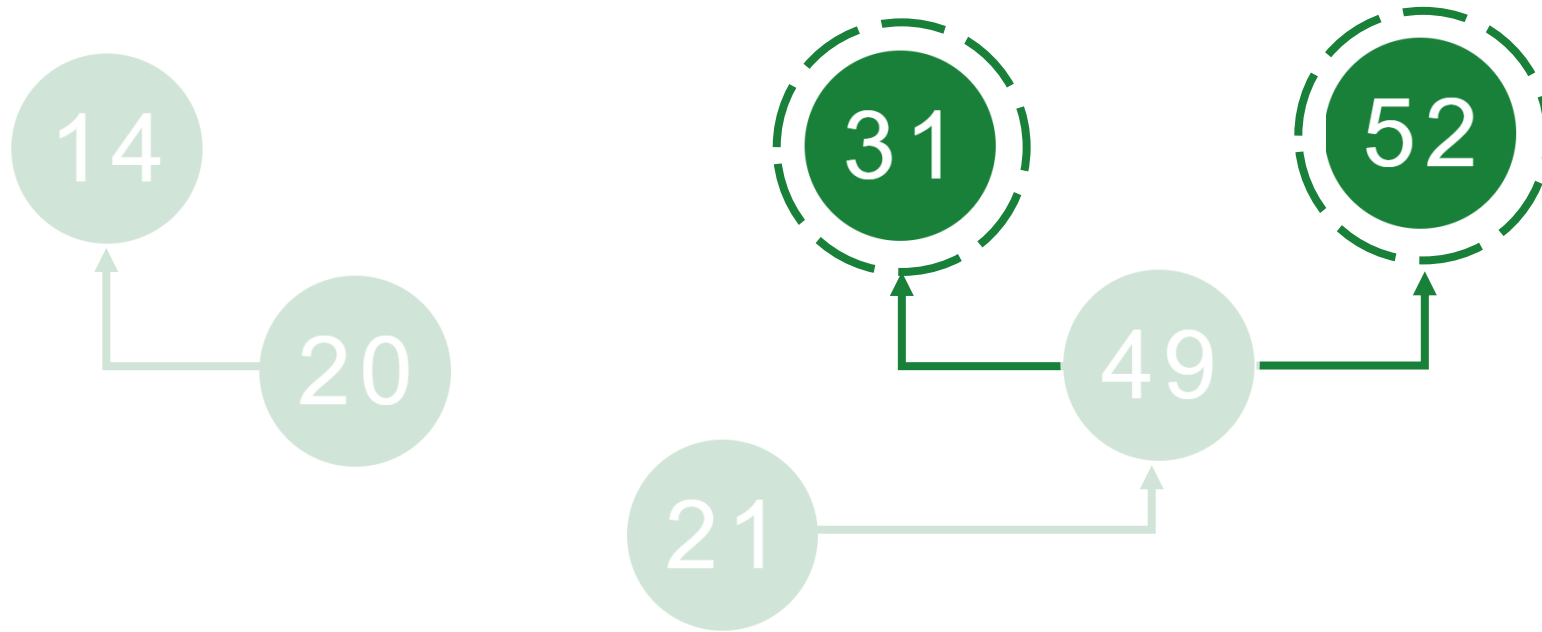
Busca Binária em Lista Lineares Encadeadas

- Busca de um elemento:



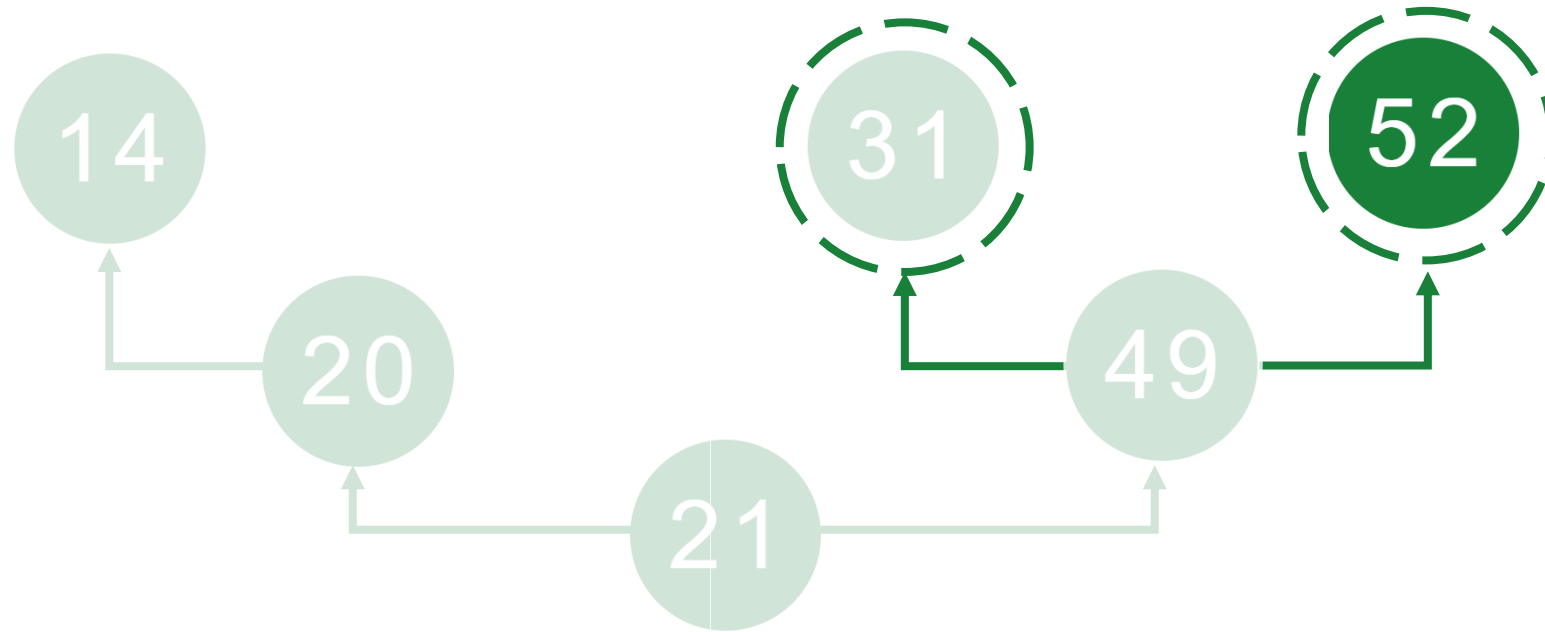
Busca Binária em Lista Lineares Encadeadas

- Busca de um elemento:



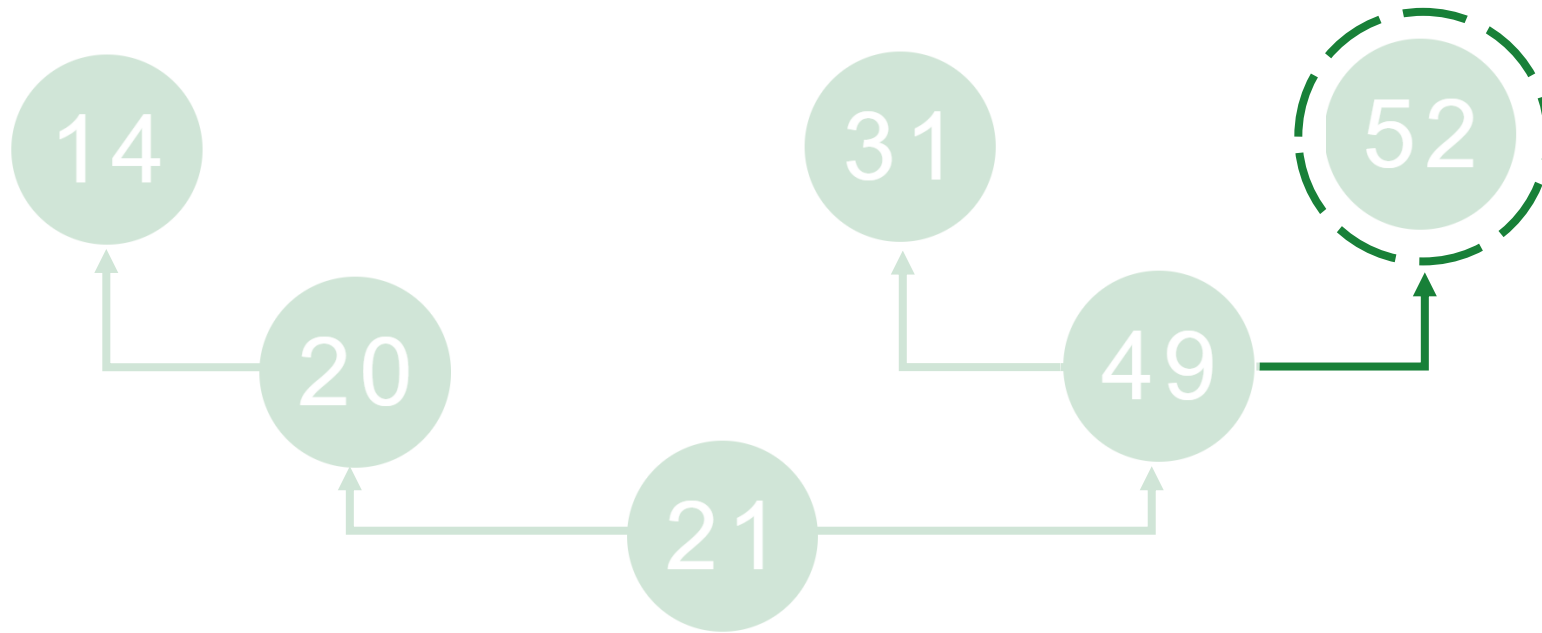
Busca Binária em Lista Lineares Encadeadas

- Busca de um elemento:



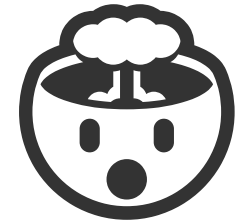
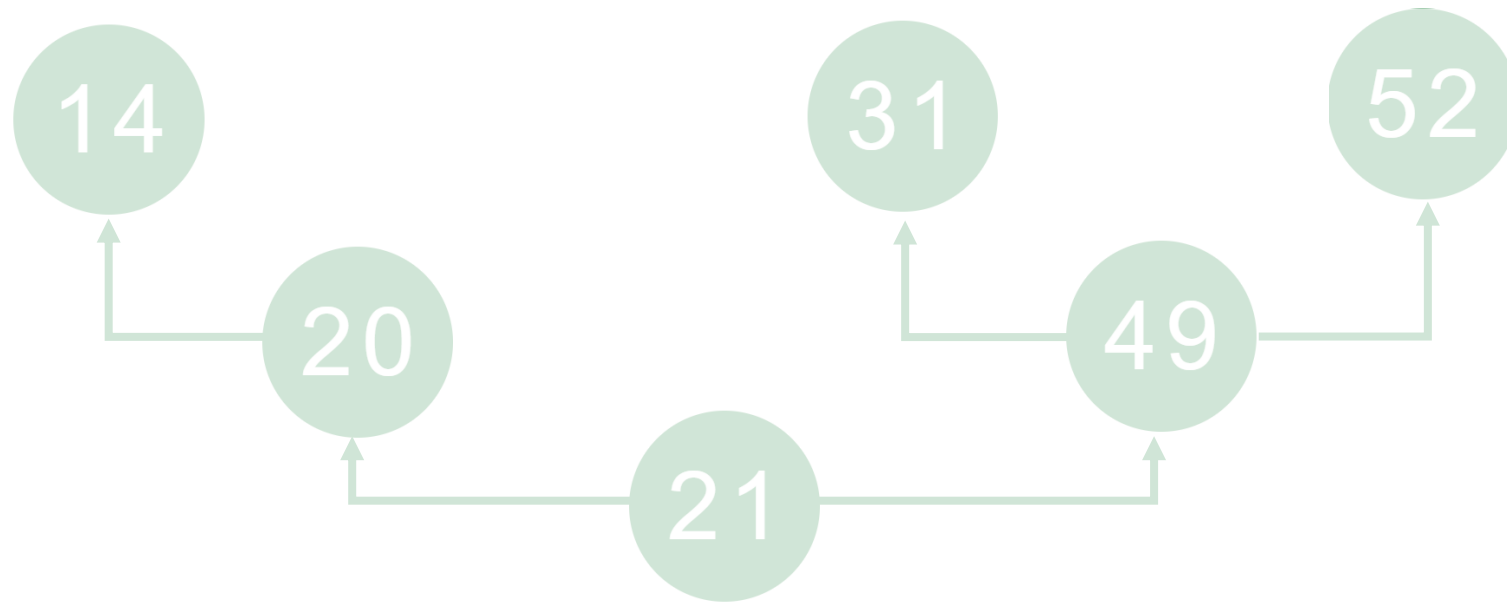
Busca Binária em Lista Lineares Encadeadas

- Busca de um elemento:



Busca Binária em Lista Lineares Encadeadas

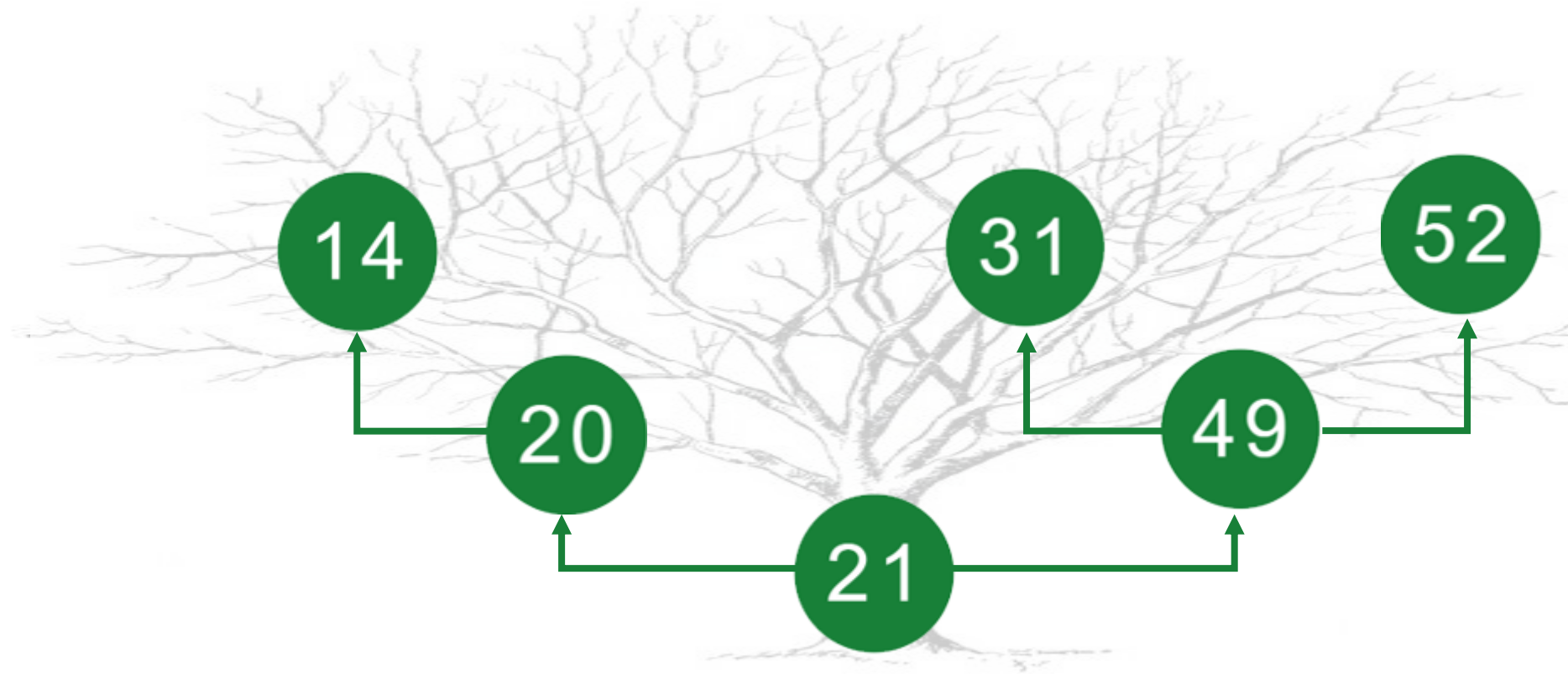
- Busca de um elemento:



Estrutura que
possibilita a
Busca Binária
com **Lista**
Encadeada!



Árvores



Estrutura que
possibilita a
Busca Binária
com **Lista**
Encadeada!

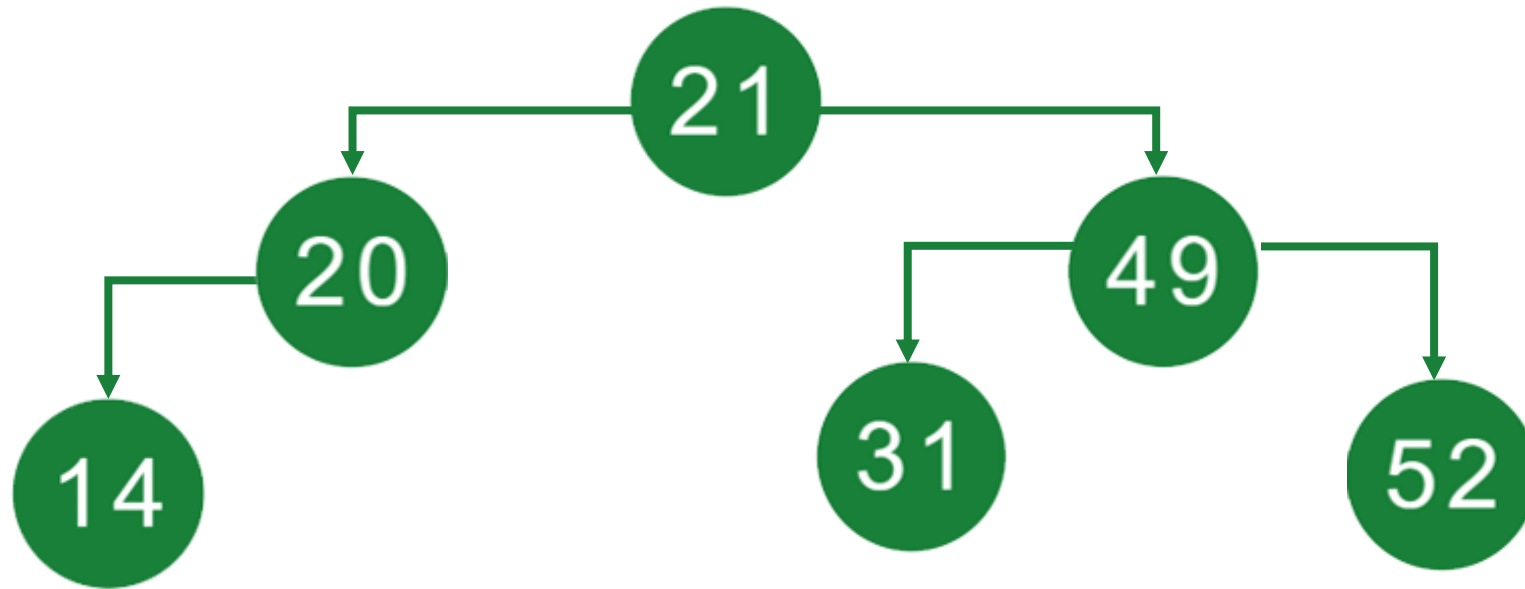
Árvores

- Estrutura de Dados **Não-Linear**.
- Sou **topologia** é baseada em **árvore**.
- Os **elementos** ficam **dispostos** de forma **hierárquica**.



Árvores

- Em **computação** representamos uma **Árvore** de forma **invertida**:

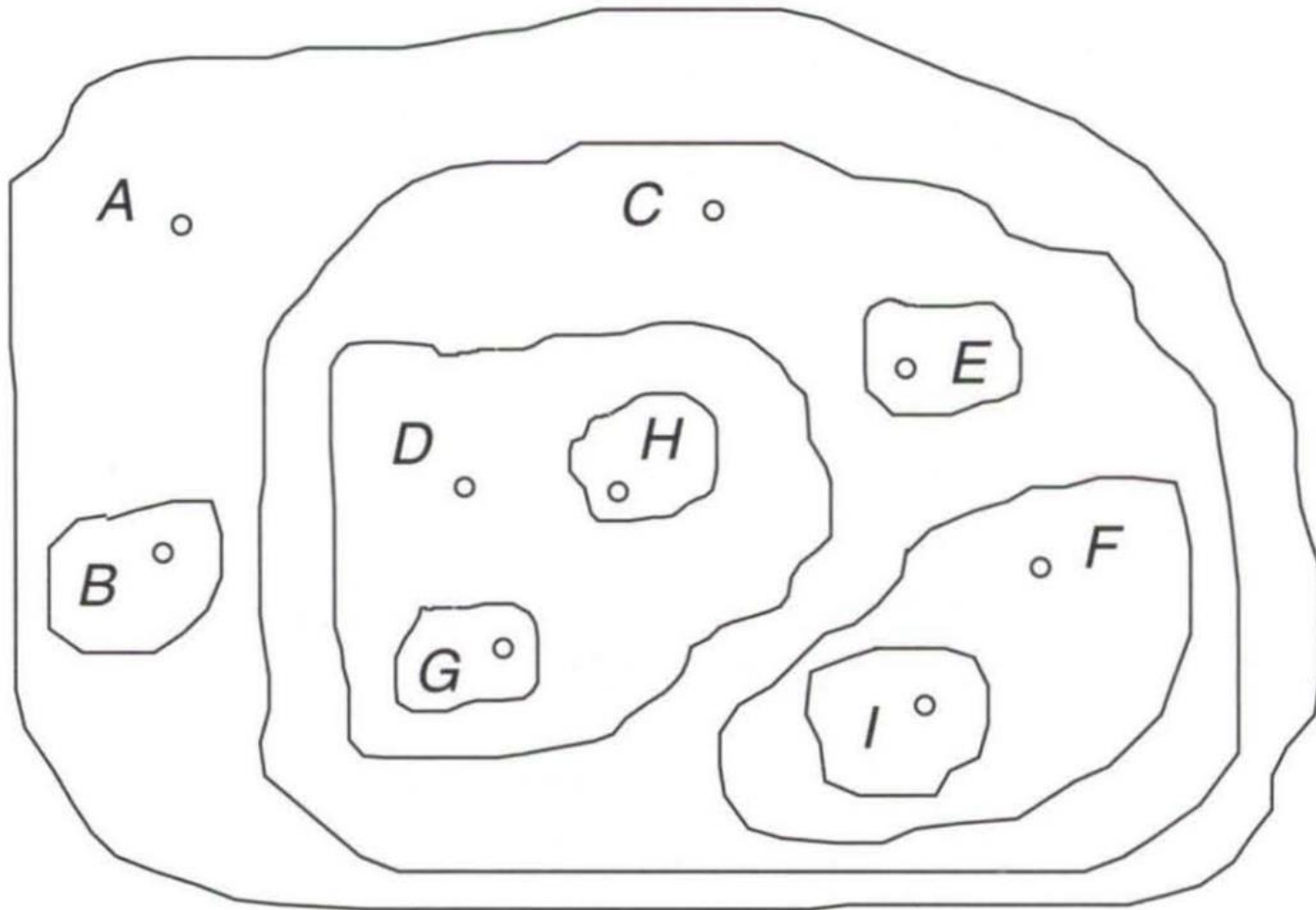



Árvores

- Uma árvore T é um conjunto finito de elementos denominados **nós**, tal que:
 - $T = 0$, árvore vazia, ou
 - $T > 0$
 - Existe um **nó** especial r denominado **raiz** de T .
 - Os nós restantes (**nós filhos**) constituem um único conjunto vazio ou formam $m \geq 1$ conjuntos disjuntos não vazios, denominados **subárvores** de r .



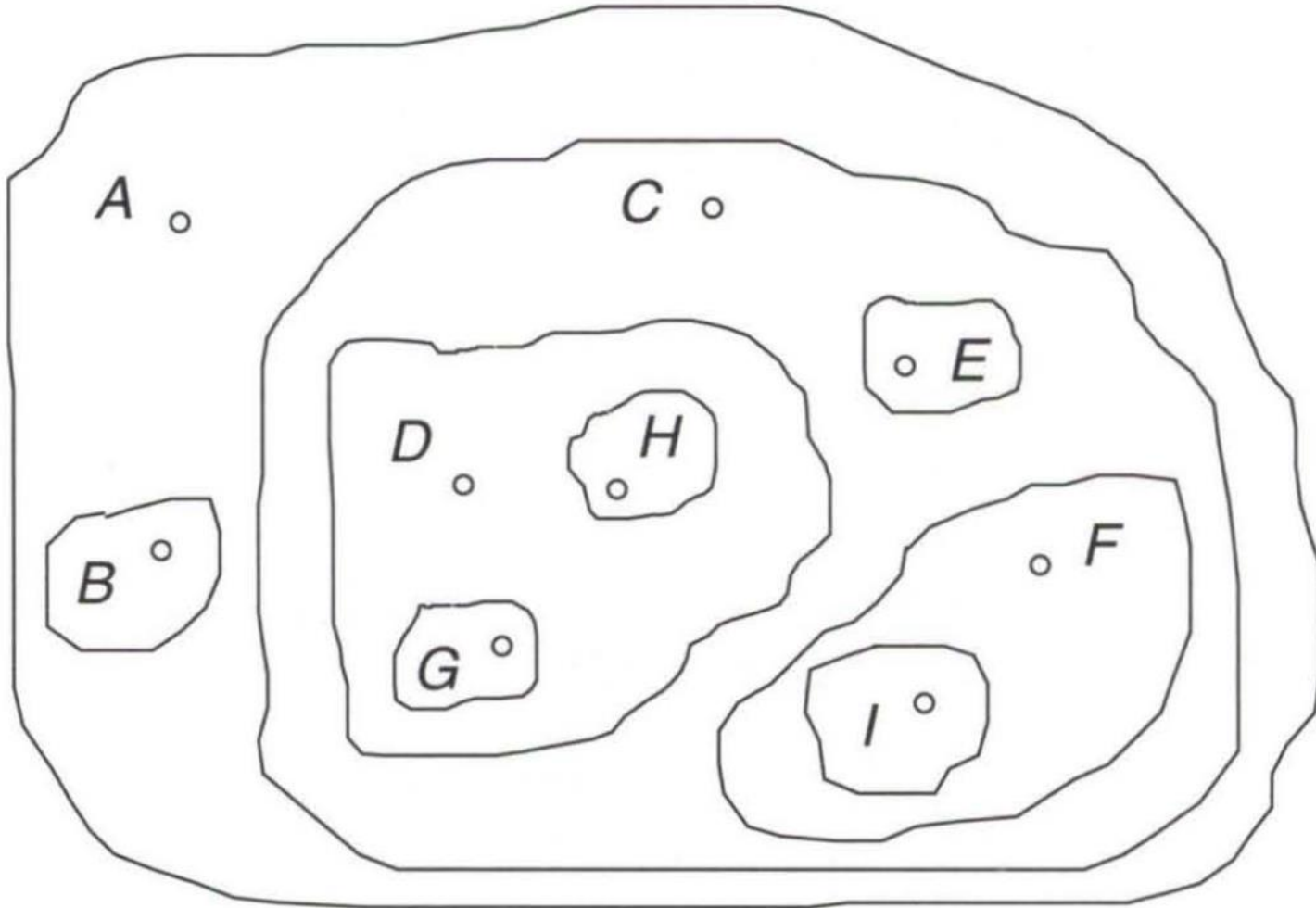
Representação de uma Árvore




-  Diagrama de Inclusão.



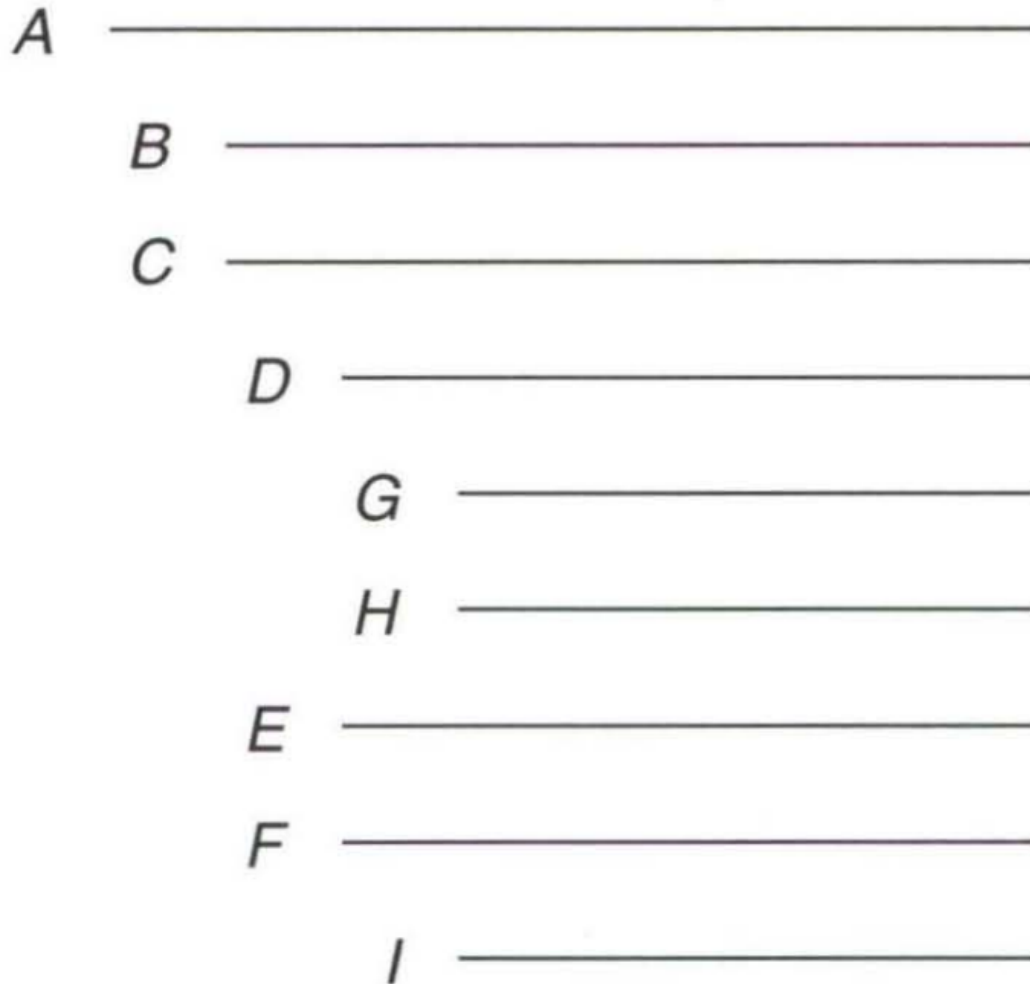
Representação de uma Árvore




-  Transforme em uma Representação Hierárquica - **Árvore.**



Representação de uma Árvore



-  Diagrama de Barras.

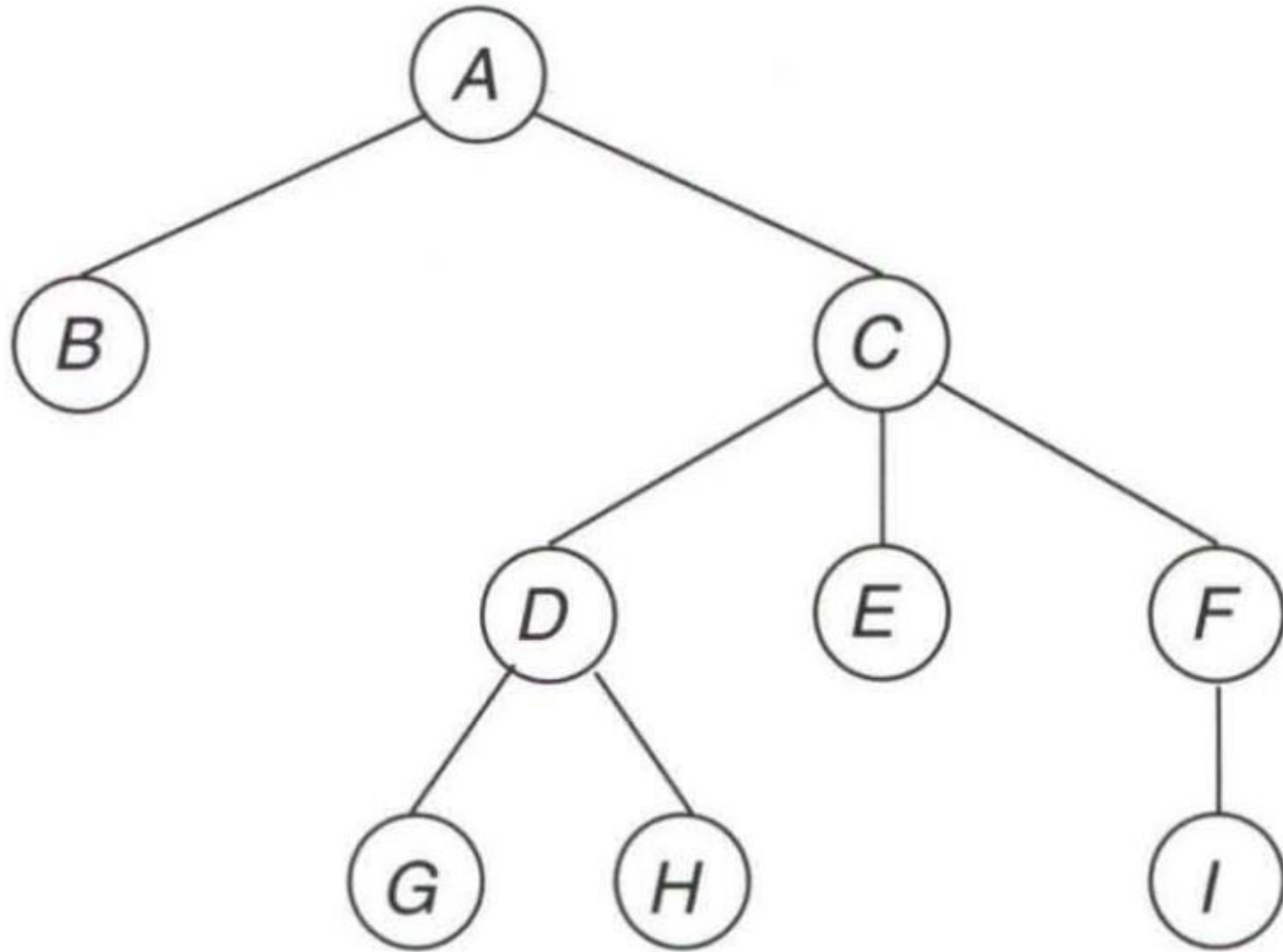


Representação de uma Árvore

(A (B) (C (D (G) (H)) (E) (F (I)))) •  Notação de Parênteses.



Representação de uma Árvore



-  Representação Hierárquica - **mais empregada.**



Elementos de uma Árvore

- **Nó:**

- Elemento que **contém** a **informação**.

- **Arco:**

- **Liga** dois nós.

- **Pai:**

- Nó **superior** de um arco.



Elementos de uma Árvore

- **Filho:**

- Nó **inferior** de um arco.

- **Raiz:**

- Nó **topo**, não possui ancestrais.

- **Folhas:**

- Nós das **extremidades inferiores**.
- Não têm nós filhos.



Elementos de uma Árvore

- **Grau:**

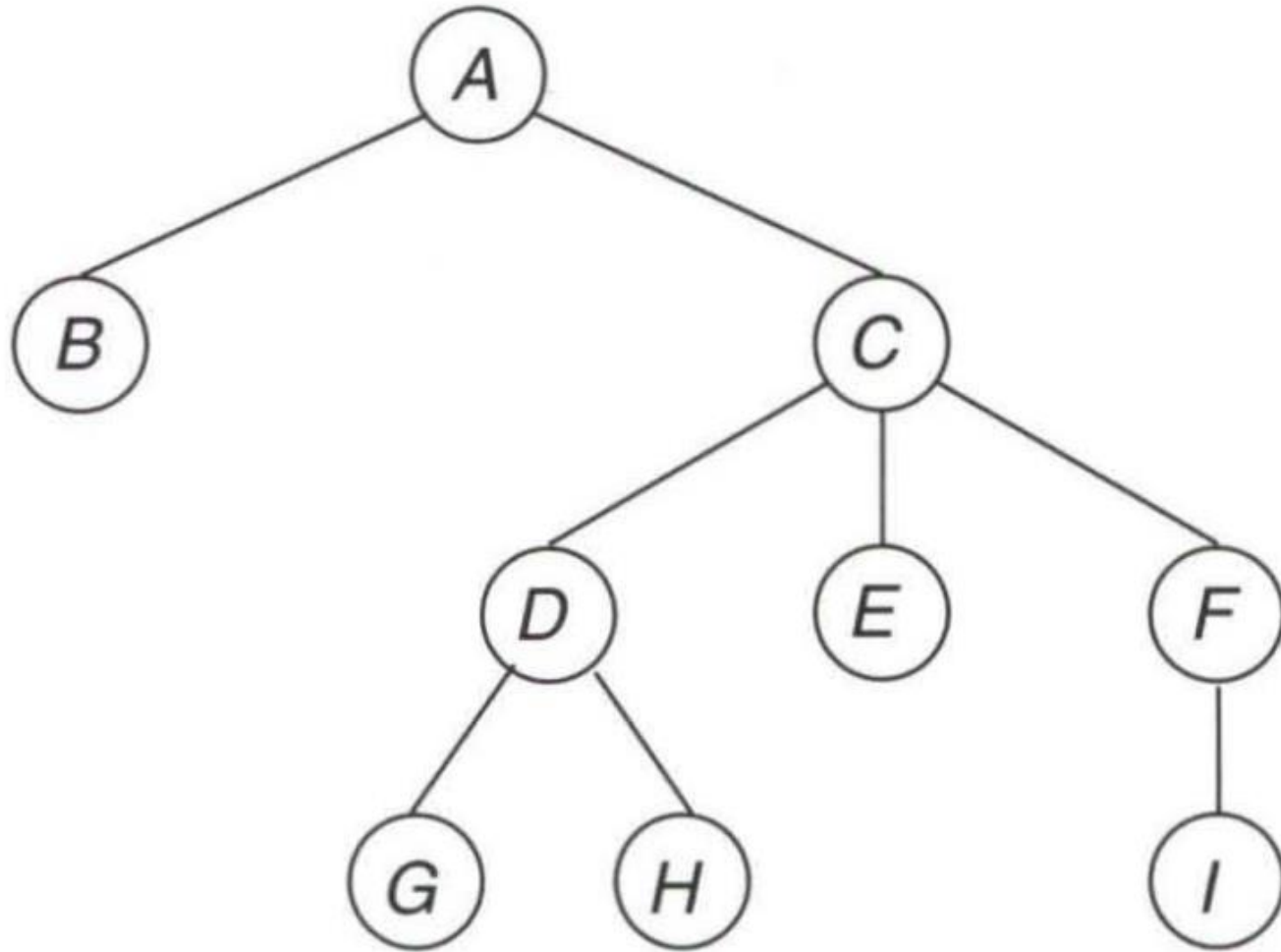
- Representa o número de subárvores de um nó.

- **Grau de uma árvore:**

- Definido como sendo igual ao máximo dos graus de todos os seus nós.



Elementos de uma Árvore

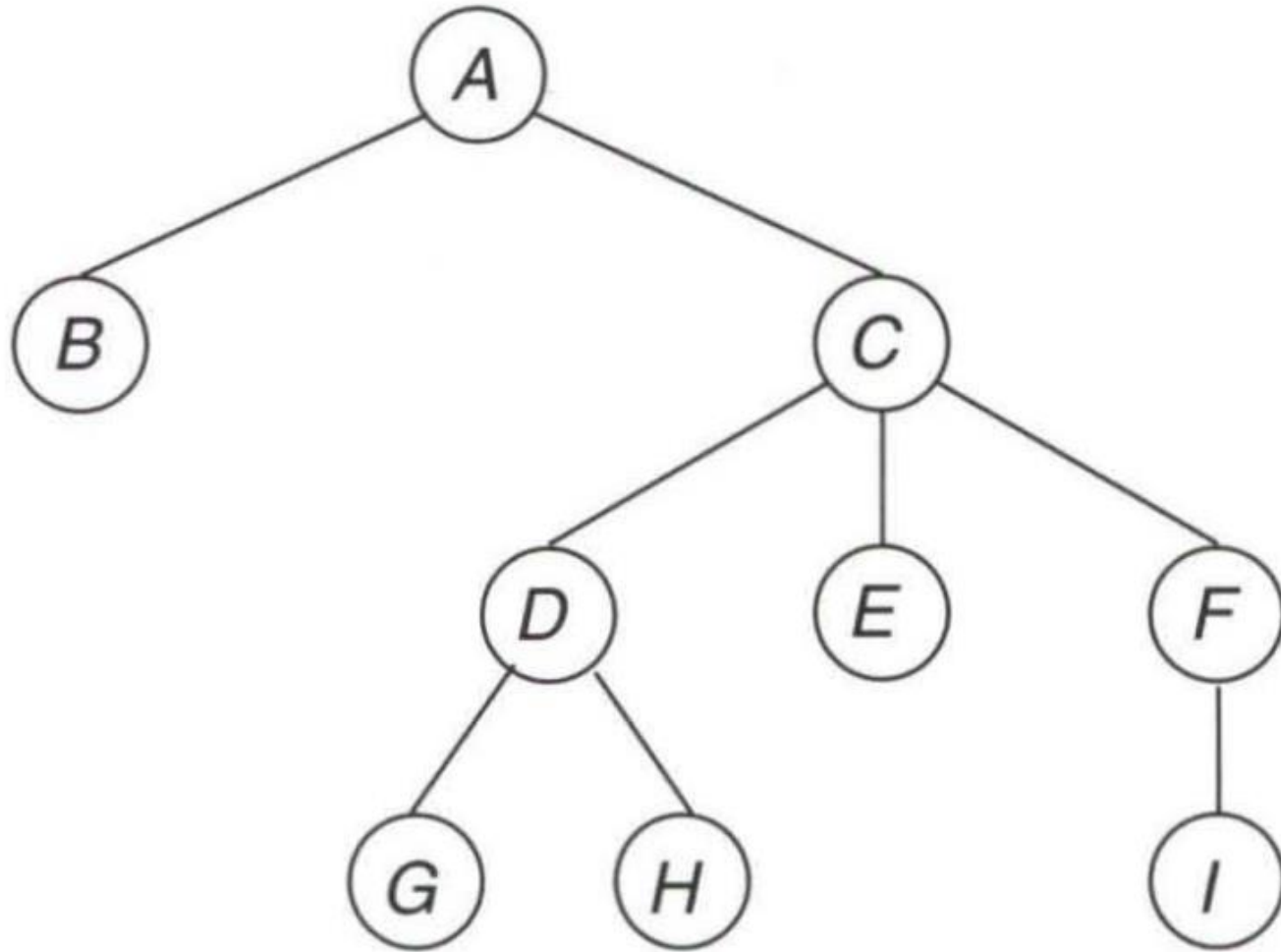


Graus dos nós

- $G(A) =$
- $G(B) =$
- $G(C) =$
- $G(D) =$
- $G(E) =$
- $G(F) =$
- $G(G) =$
- $G(H) =$
- $G(I) =$
- $\text{Grau}(T) =$



Elementos de uma Árvore

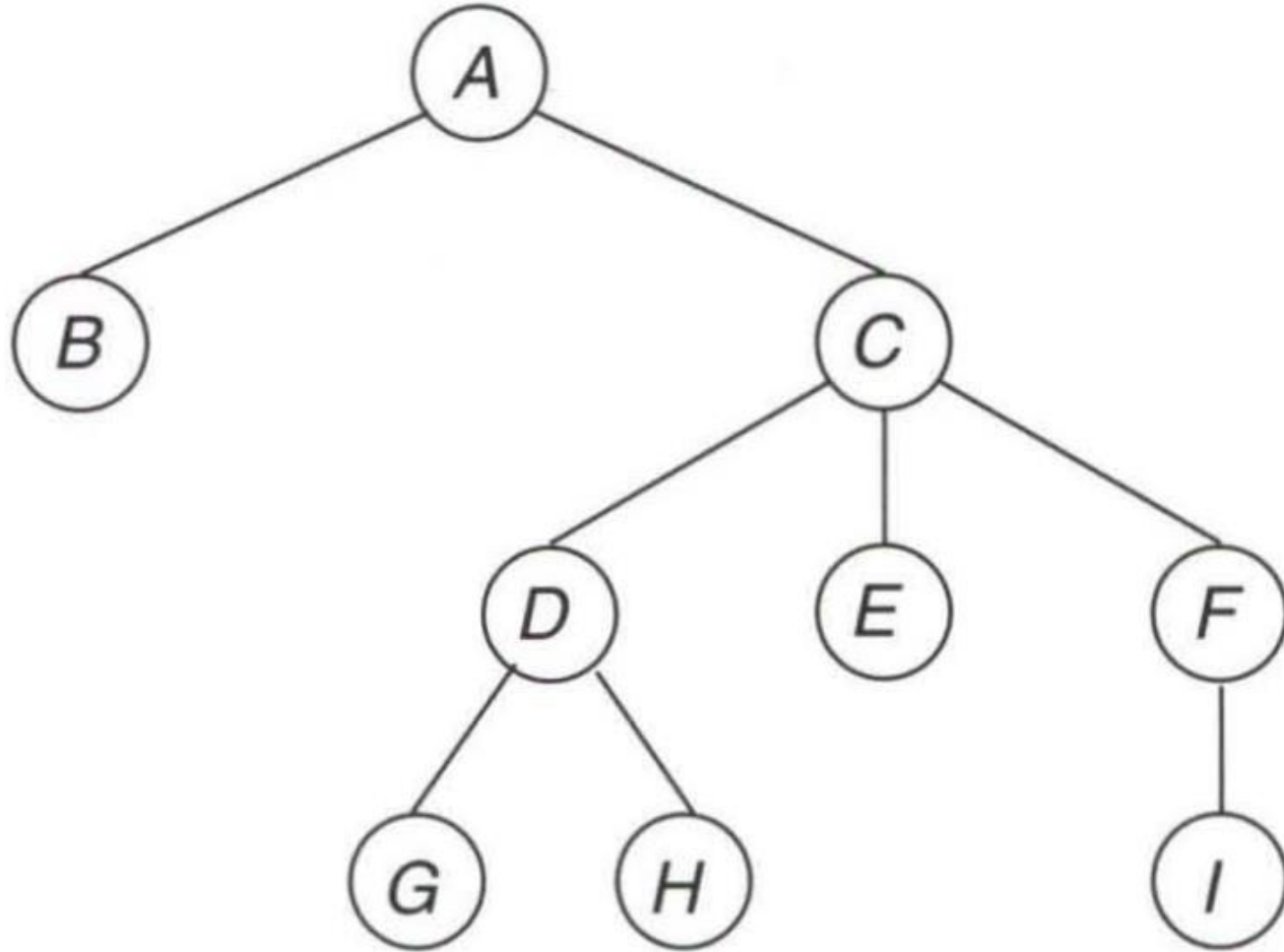


Graus dos nós

- $G(A) = 2$
- $G(B) =$
- $G(C) =$
- $G(D) =$
- $G(E) =$
- $G(F) =$
- $G(G) =$
- $G(H) =$
- $G(I) =$
- $\text{Grau}(T) =$



Elementos de uma Árvore

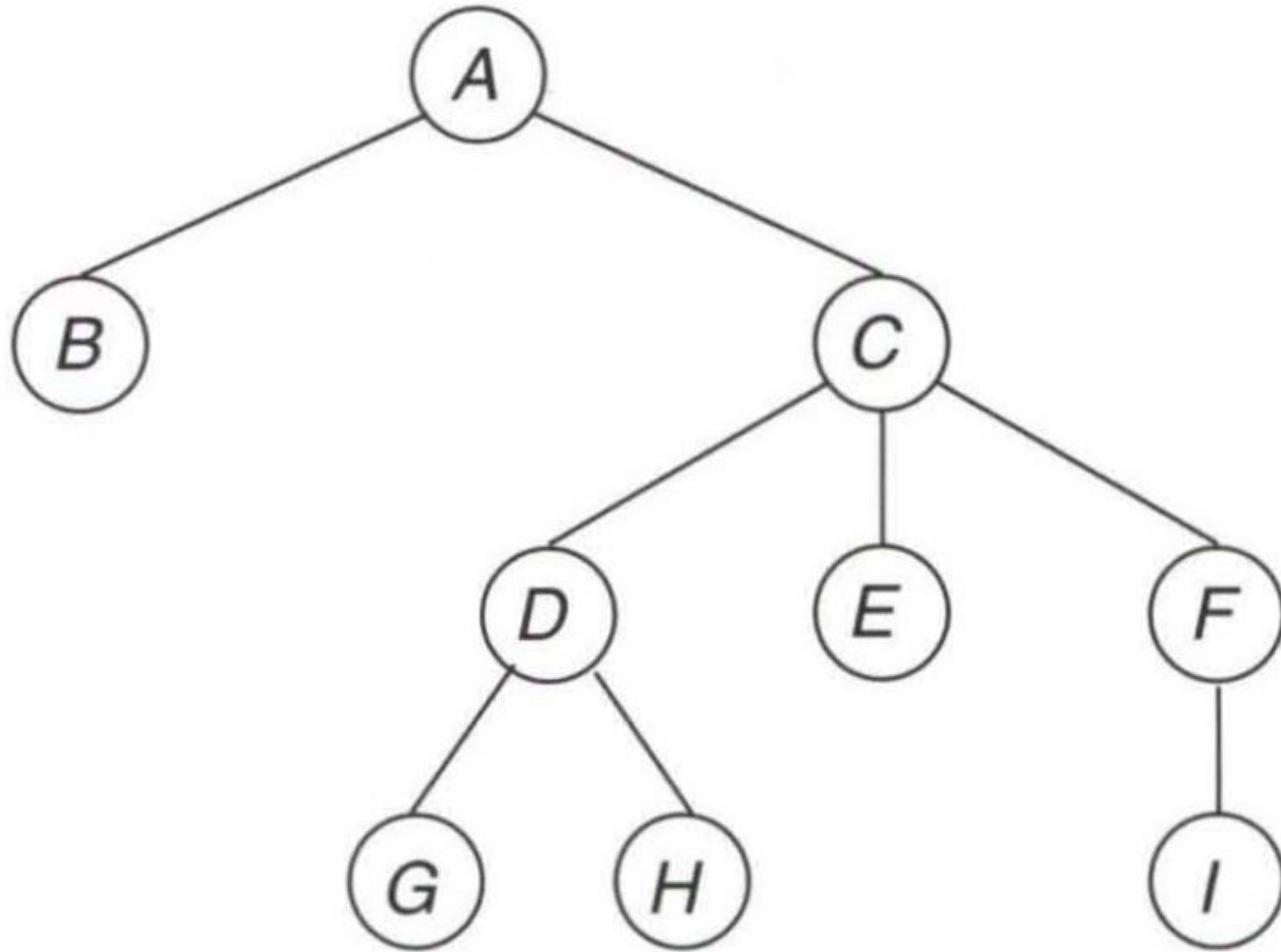


Graus dos nós

- $G(A) = 2$
- $G(B) = 0$
- $G(C) =$
- $G(D) =$
- $G(E) =$
- $G(F) =$
- $G(G) =$
- $G(H) =$
- $G(I) =$
- $\text{Grau}(T) =$



Elementos de uma Árvore

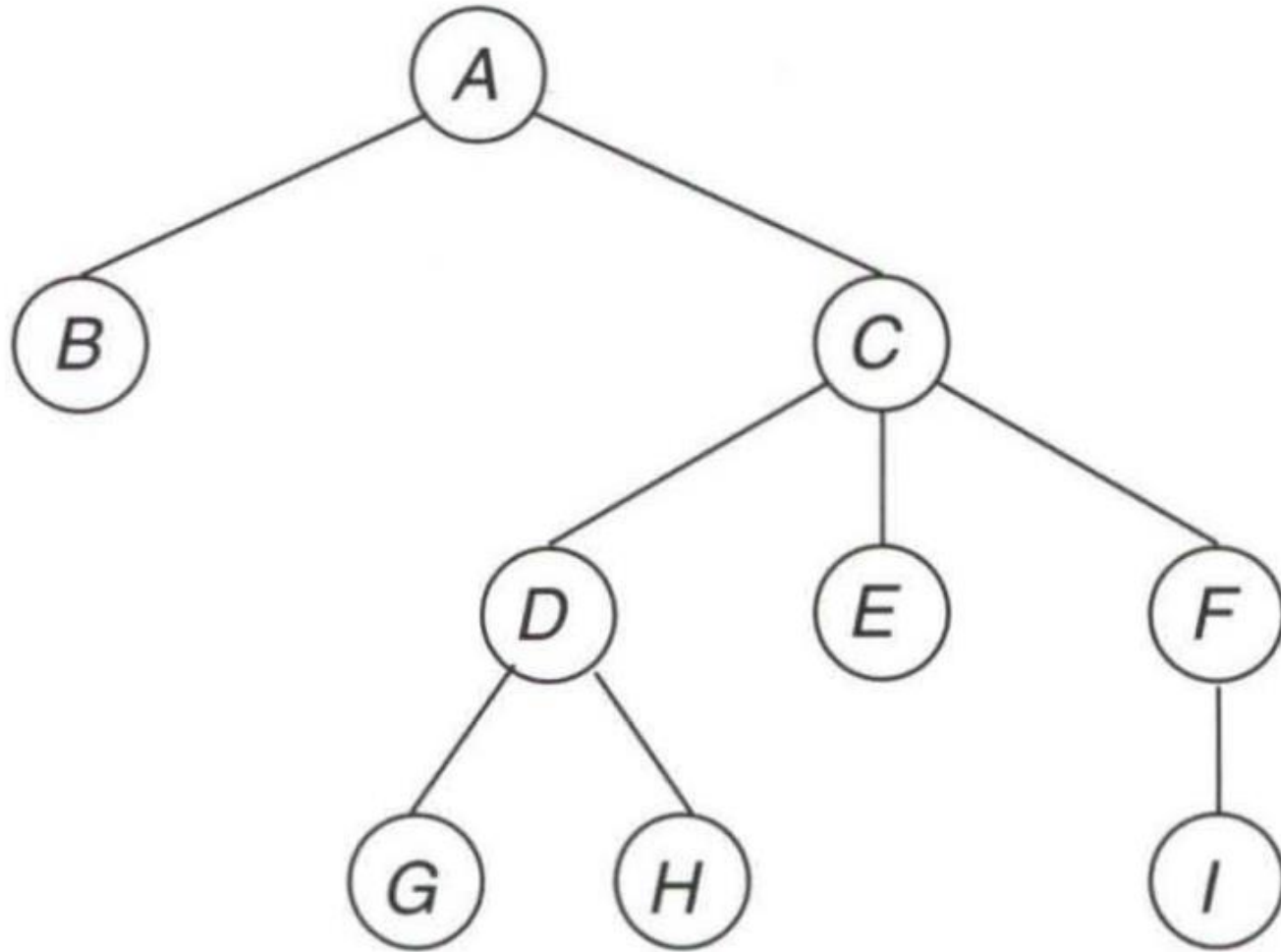


Graus dos nós

- $G(A) = 2$
- $G(B) = 0$
- $G(C) = 3$
- $G(D) =$
- $G(E) =$
- $G(F) =$
- $G(G) =$
- $G(H) =$
- $G(I) =$
- $\text{Grau}(T) =$



Elementos de uma Árvore

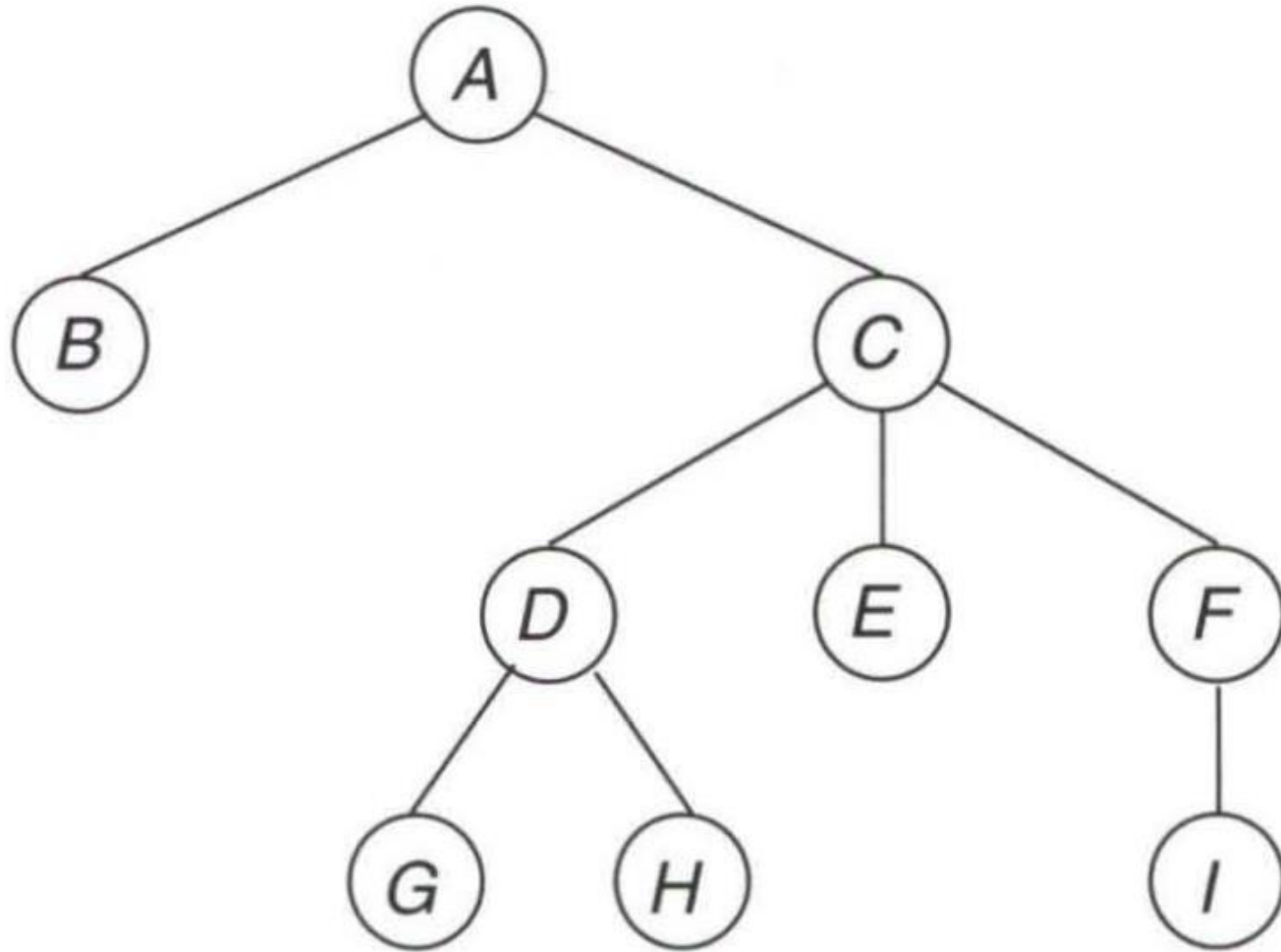


Graus dos nós

- $G(A) = 2$
- $G(B) = 0$
- $G(C) = 3$
- $G(D) = 2$
- $G(E) =$
- $G(F) =$
- $G(G) =$
- $G(H) =$
- $G(I) =$
- $\text{Grau}(T) =$



Elementos de uma Árvore

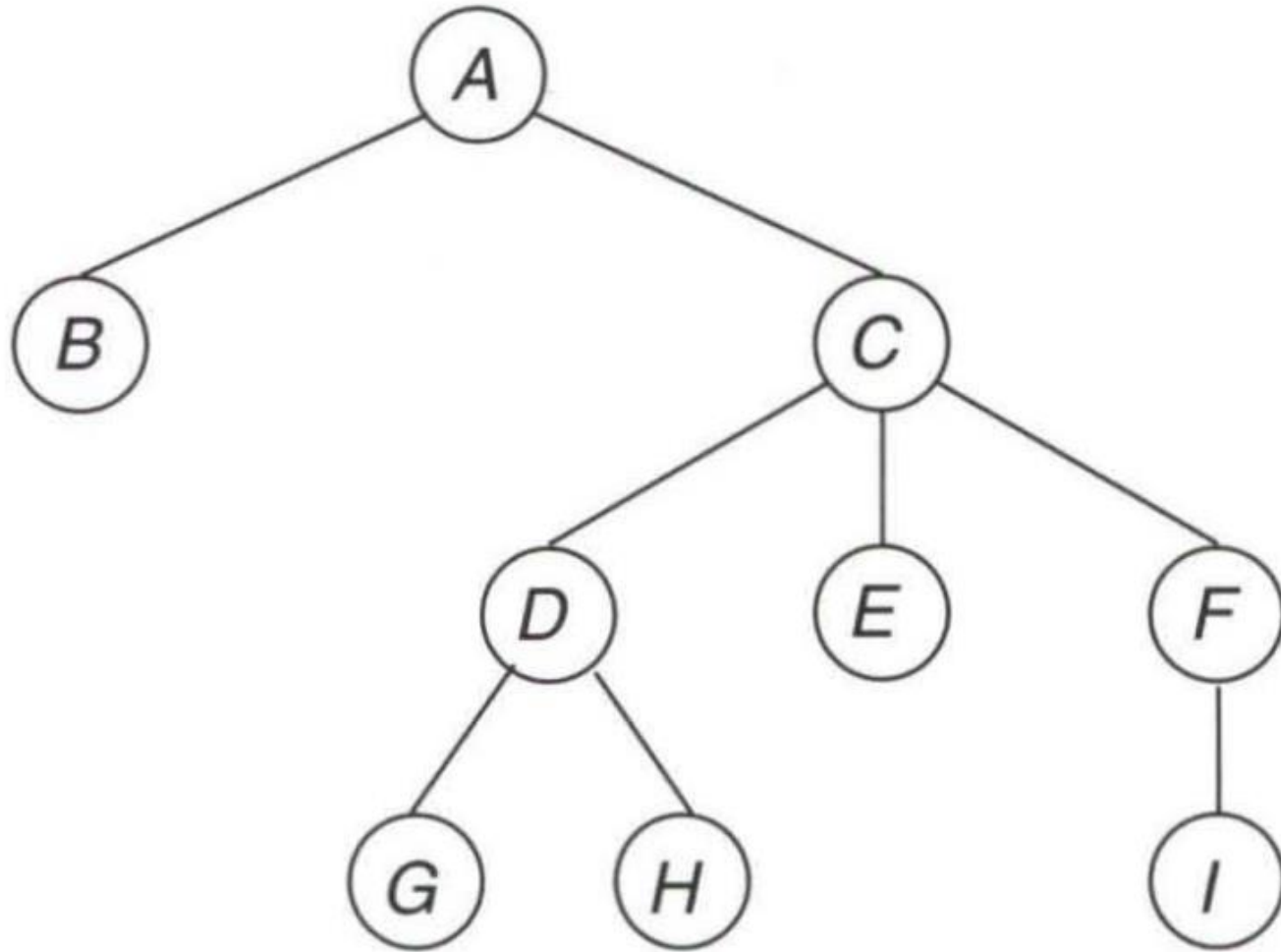


Graus dos nós

- $G(A) = 2$
- $G(B) = 0$
- $G(C) = 3$
- $G(D) = 2$
- $G(E) = 0$
- $G(F) =$
- $G(G) =$
- $G(H) =$
- $G(I) =$
- $\text{Grau}(T) =$



Elementos de uma Árvore

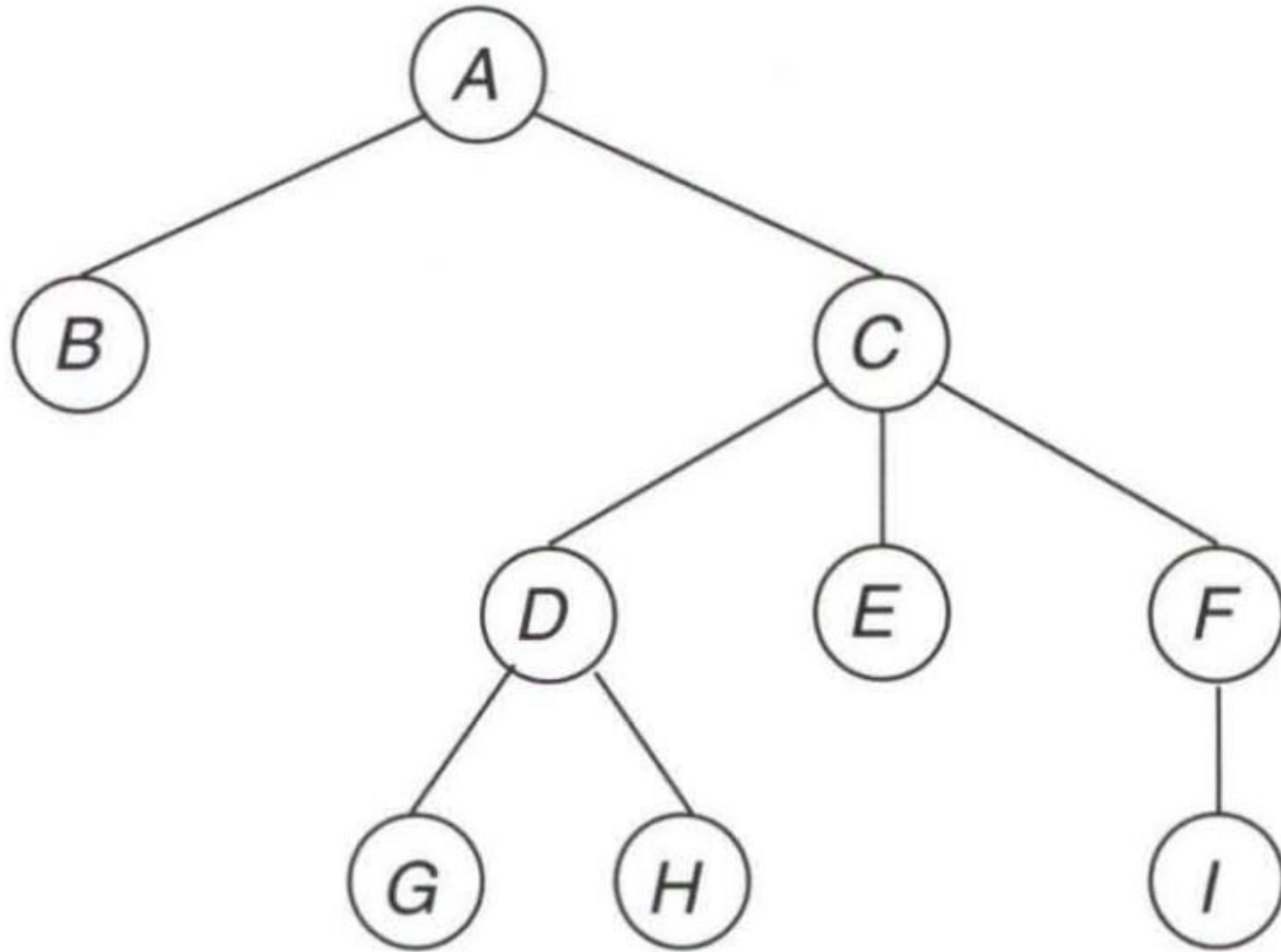


Graus dos nós

- $G(A) = 2$
- $G(B) = 0$
- $G(C) = 3$
- $G(D) = 2$
- $G(E) = 0$
- $G(F) = 1$
- $G(G) =$
- $G(H) =$
- $G(I) =$
- $\text{Grau}(T) =$



Elementos de uma Árvore

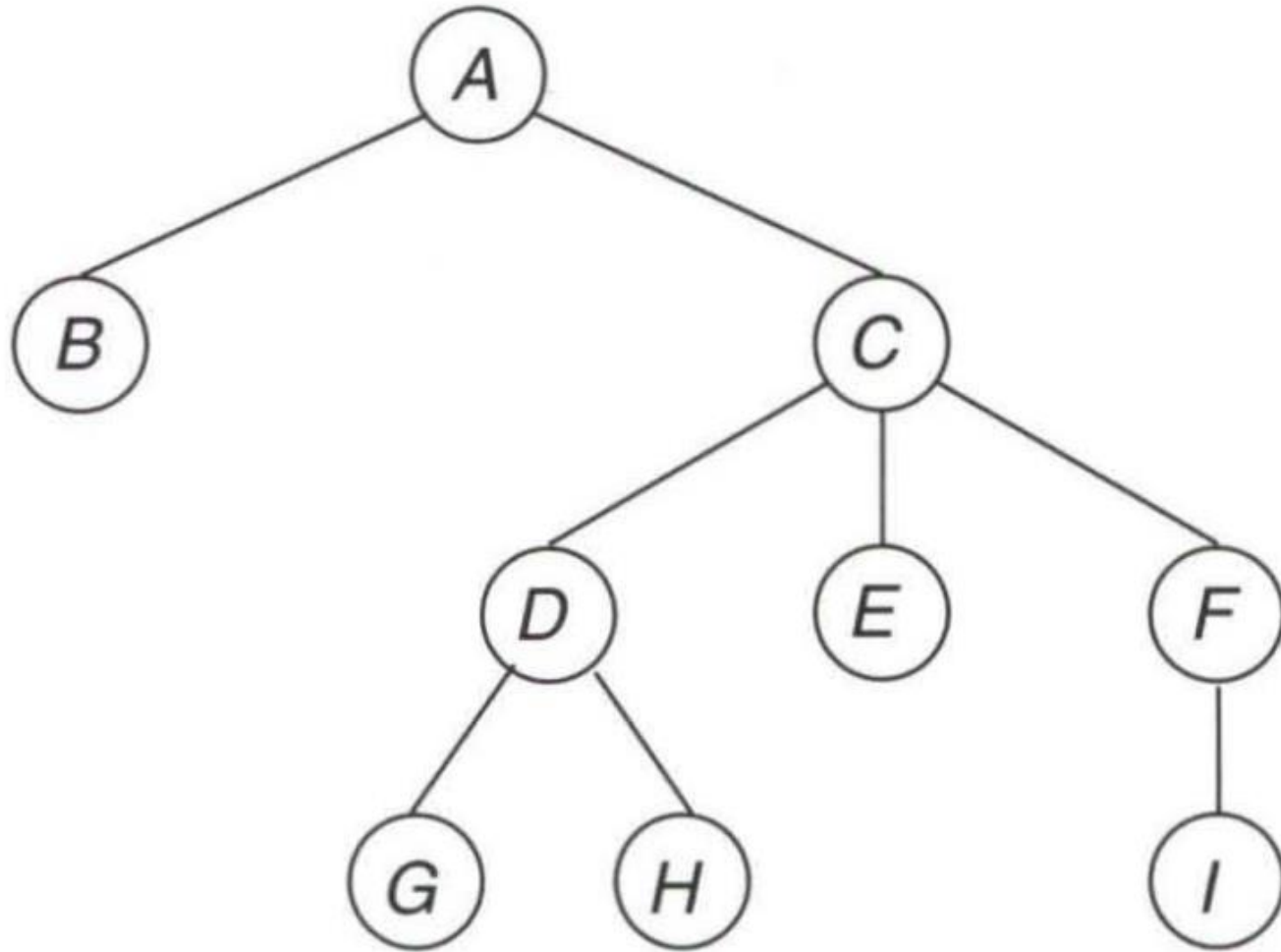


Graus dos nós

- $G(A) = 2$
- $G(B) = 0$
- $G(C) = 3$
- $G(D) = 2$
- $G(E) = 0$
- $G(F) = 1$
- $G(G) = 0$
- $G(H) =$
- $G(I) =$
- $\text{Grau}(T) =$



Elementos de uma Árvore

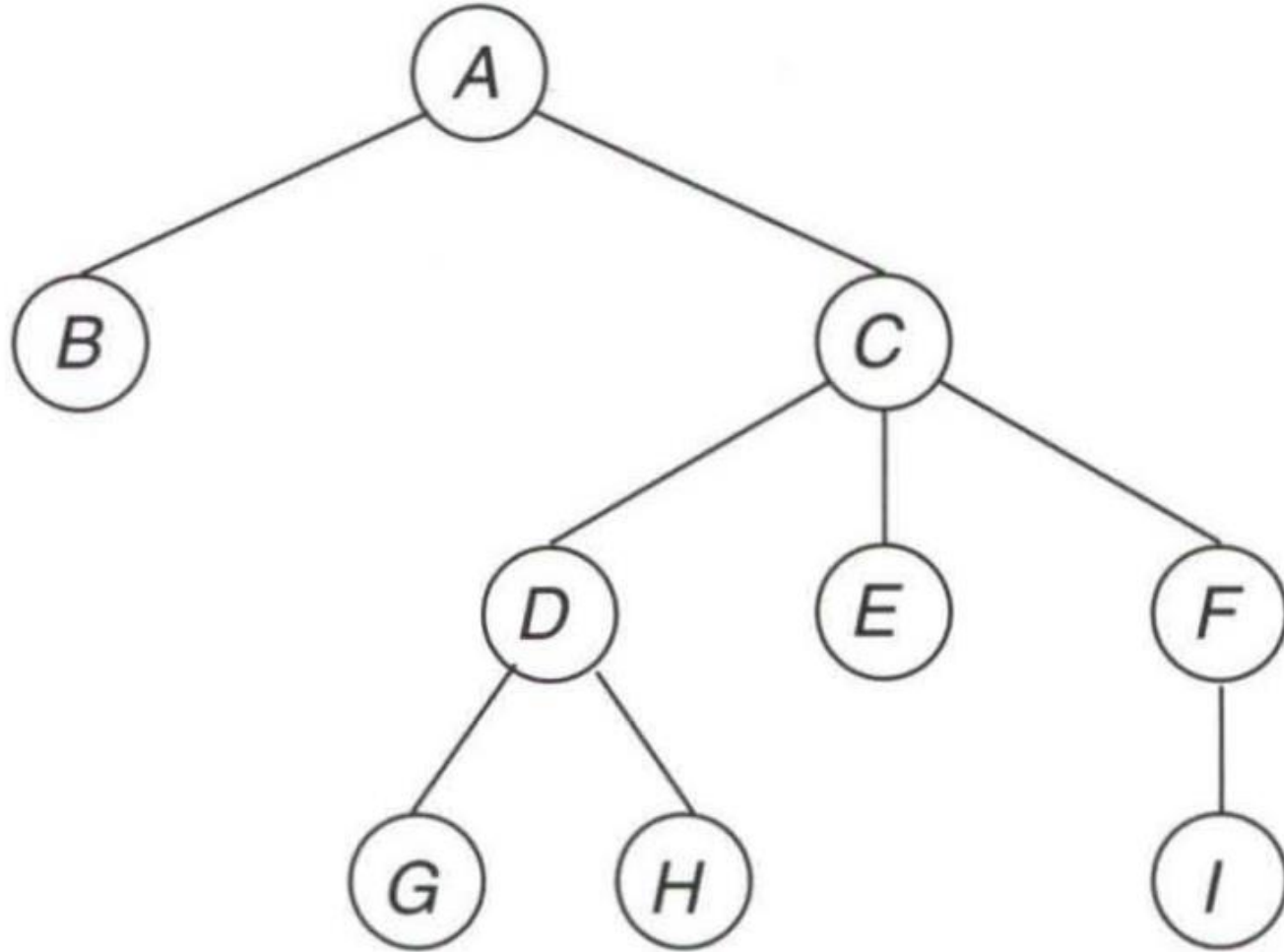


Graus dos nós

- $G(A) = 2$
- $G(B) = 0$
- $G(C) = 3$
- $G(D) = 2$
- $G(E) = 0$
- $G(F) = 1$
- $G(G) = 0$
- $G(H) = 0$
- $G(I) = 0$
- $\text{Grau}(T) = 9$



Elementos de uma Árvore

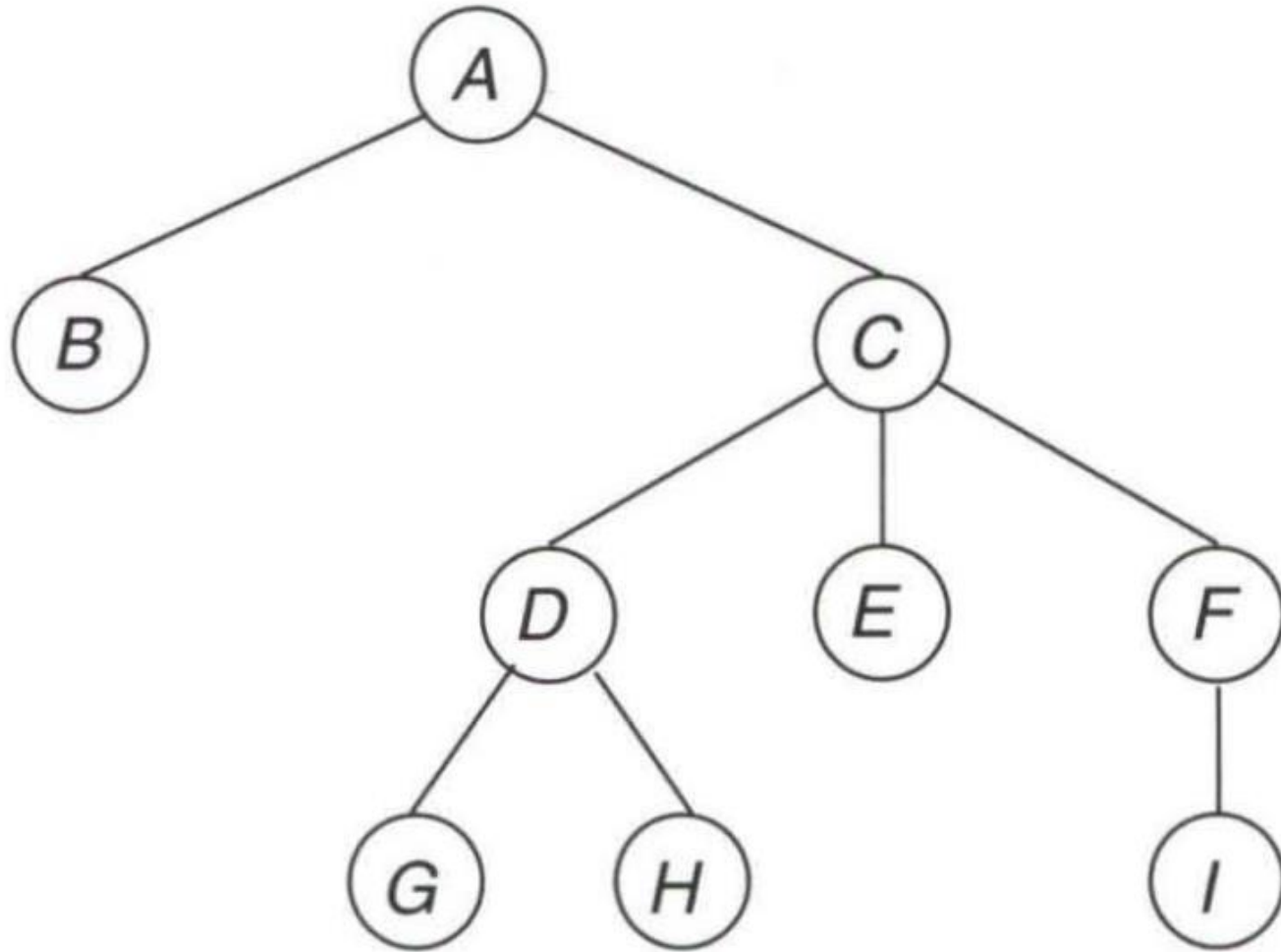


Graus dos nós

- $G(A) = 2$
- $G(B) = 0$
- $G(C) = 3$
- $G(D) = 2$
- $G(E) = 0$
- $G(F) = 1$
- $G(G) = 0$
- $G(H) = 0$
- $G(I) = 0$
- $\text{Grau}(T) =$



Elementos de uma Árvore

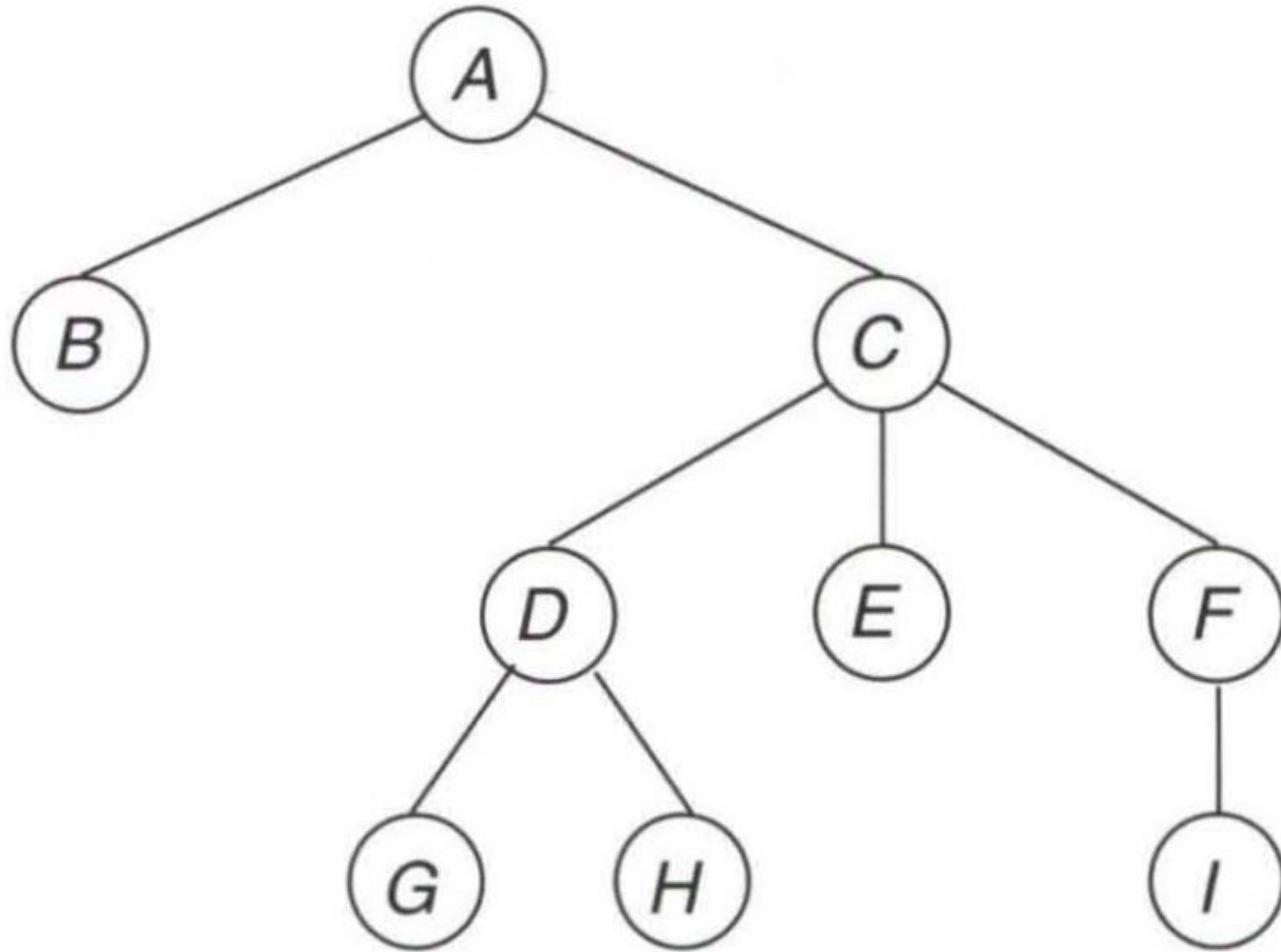


Graus dos nós

- $G(A) = 2$
- $G(B) = 0$
- $G(C) = 3$
- $G(D) = 2$
- $G(E) = 0$
- $G(F) = 1$
- $G(G) = 0$
- $G(H) = 0$
- $G(I) = 0$
- $\text{Grau}(T) = 3$



Elementos de uma Árvore



Graus dos nós

- $G(A) = 2$
- $G(B) = 0$
- $G(C) = 3$
- $G(D) = 2$
- $G(E) = 0$
- $G(F) = 1$
- $G(G) = 0$
- $G(H) = 0$
- $G(I) = 0$
- $\text{Grau}(T) = 3$



Elementos de uma Árvore

- **Caminho:**

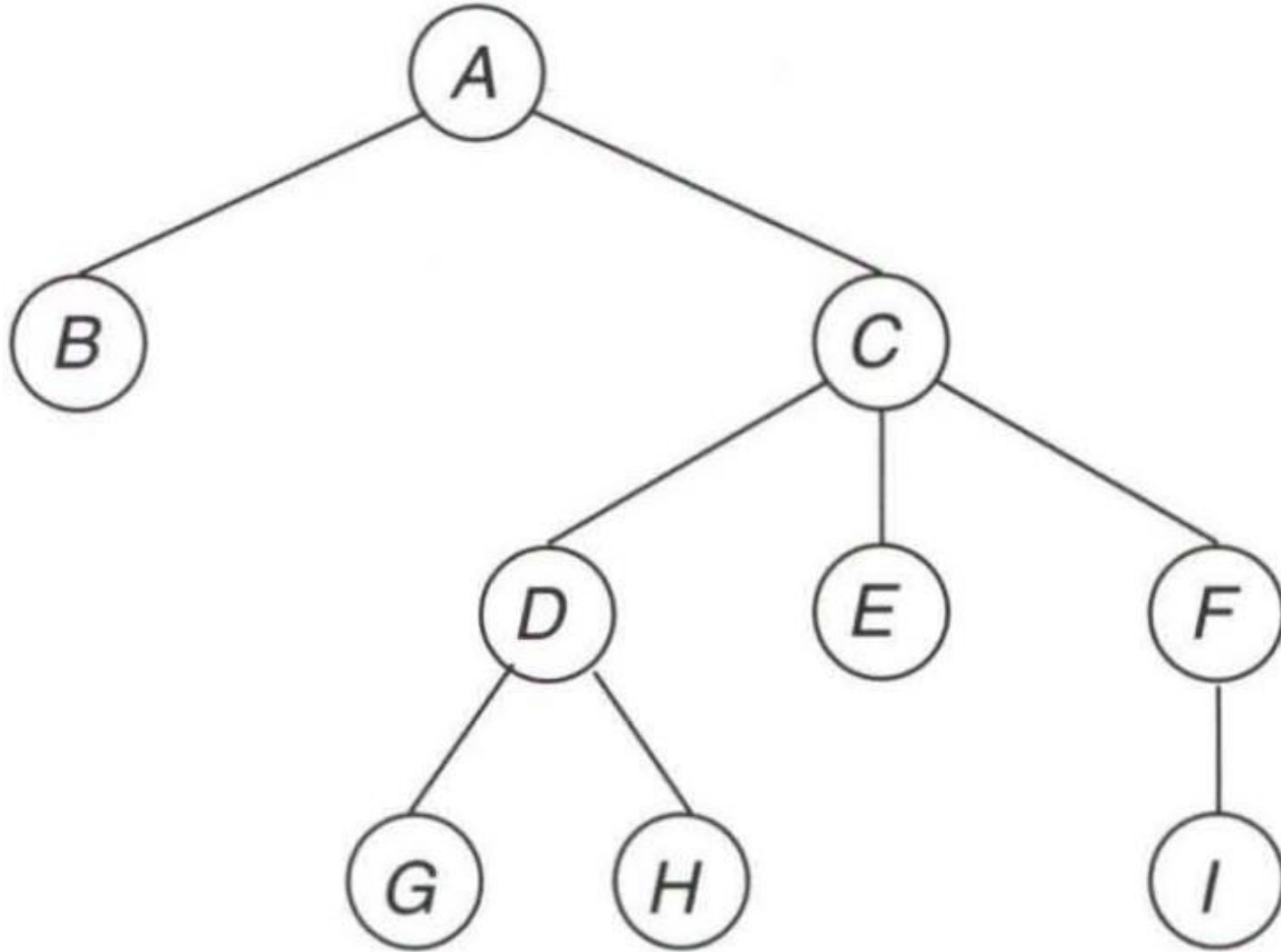
- Cada nó tem que ser **atingível** a **partir** da **raiz** através de uma **sequência única** de **arcos**.

- **Comprimento do caminho:**

- O **número** de **arcos** do **caminho**.



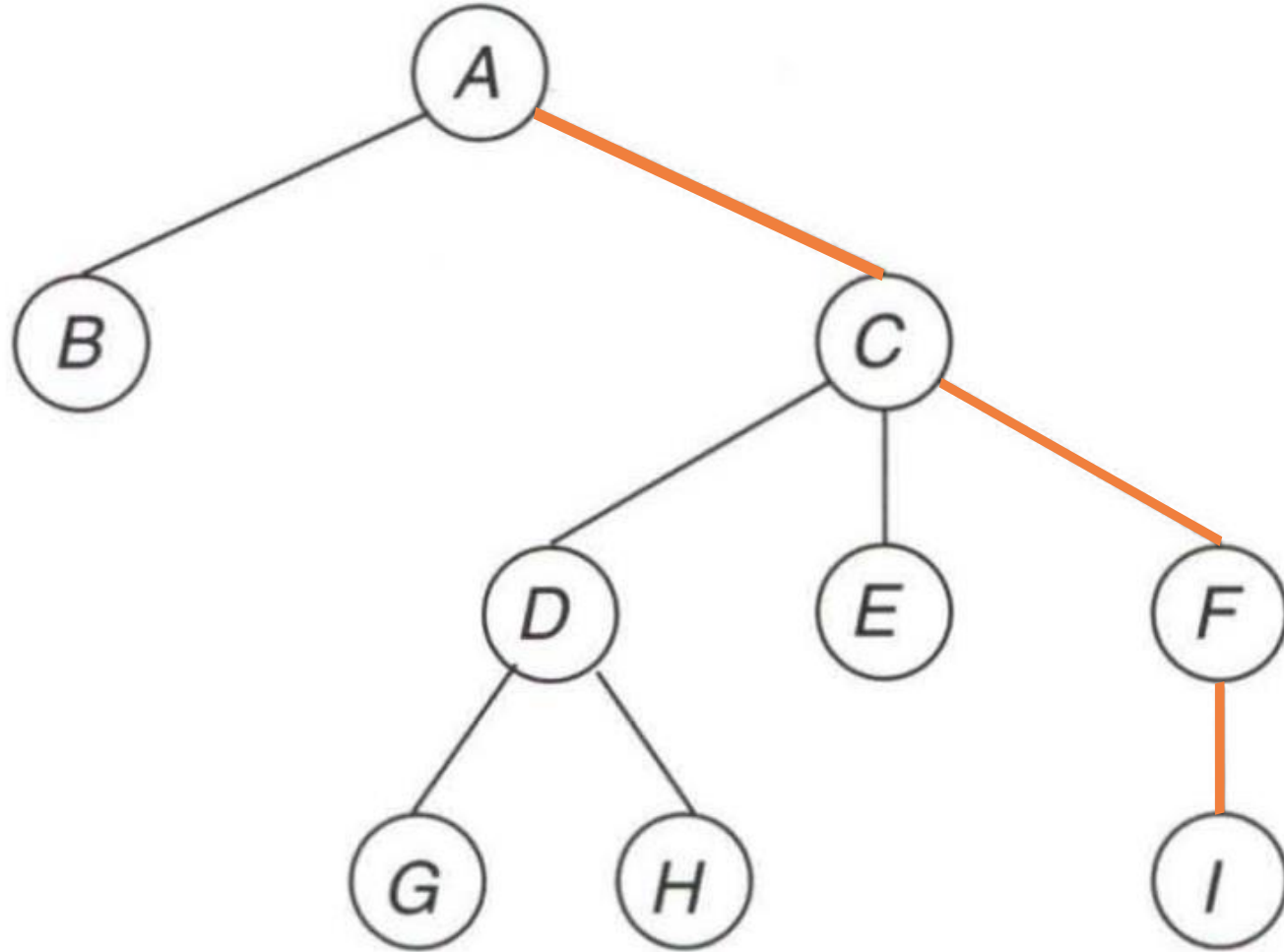
Elementos de uma Árvore



- O Caminho do nó **A** até o nó **I** tem comprimento =



Elementos de uma Árvore



- O Caminho do nó **A** até o nó **I** tem comprimento = **3**.

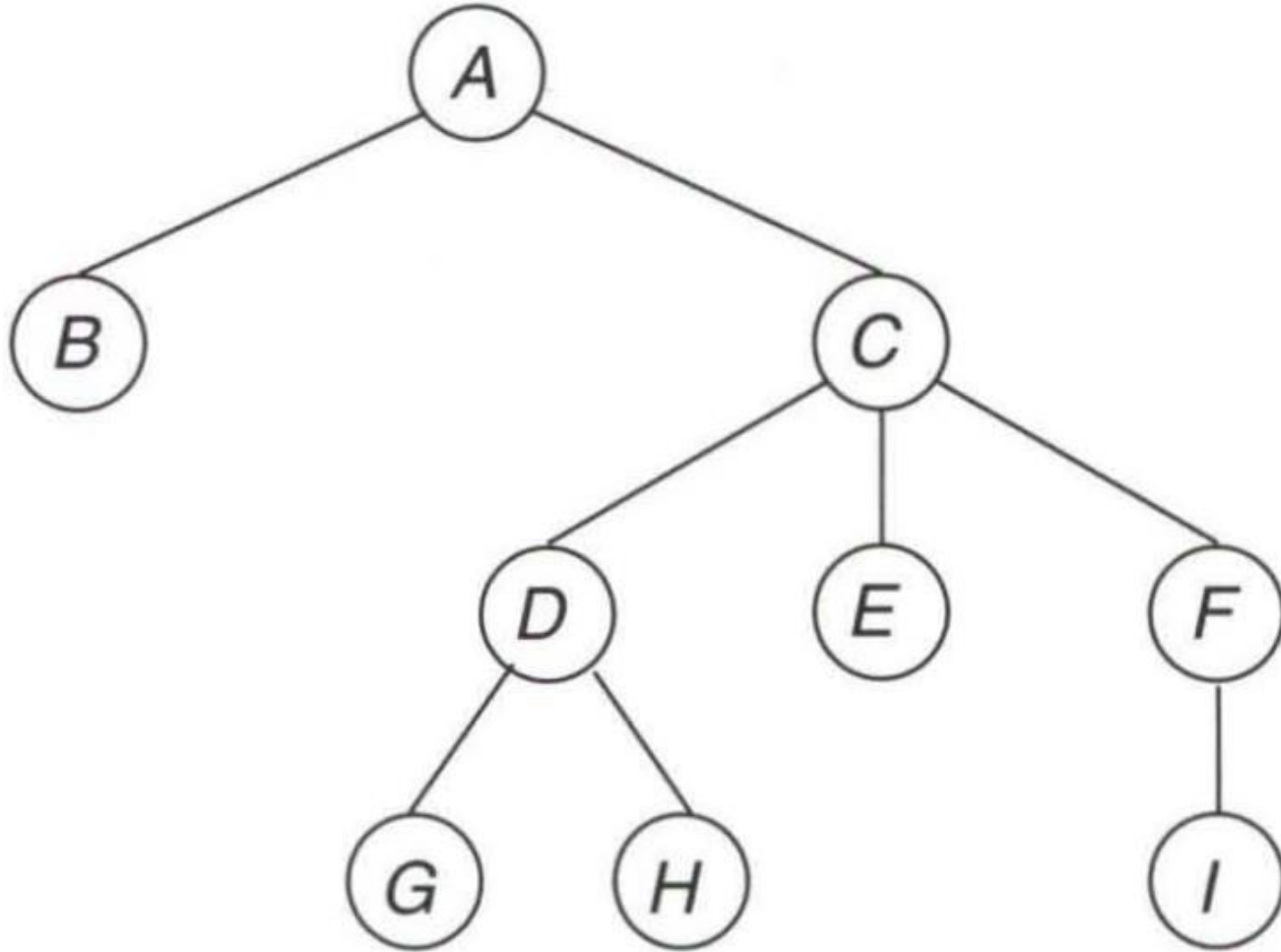


Elementos de uma Árvore

- **Nível de um nó:**
 - É a **distância** de um **nó** até a **raiz** da **Árvore**.

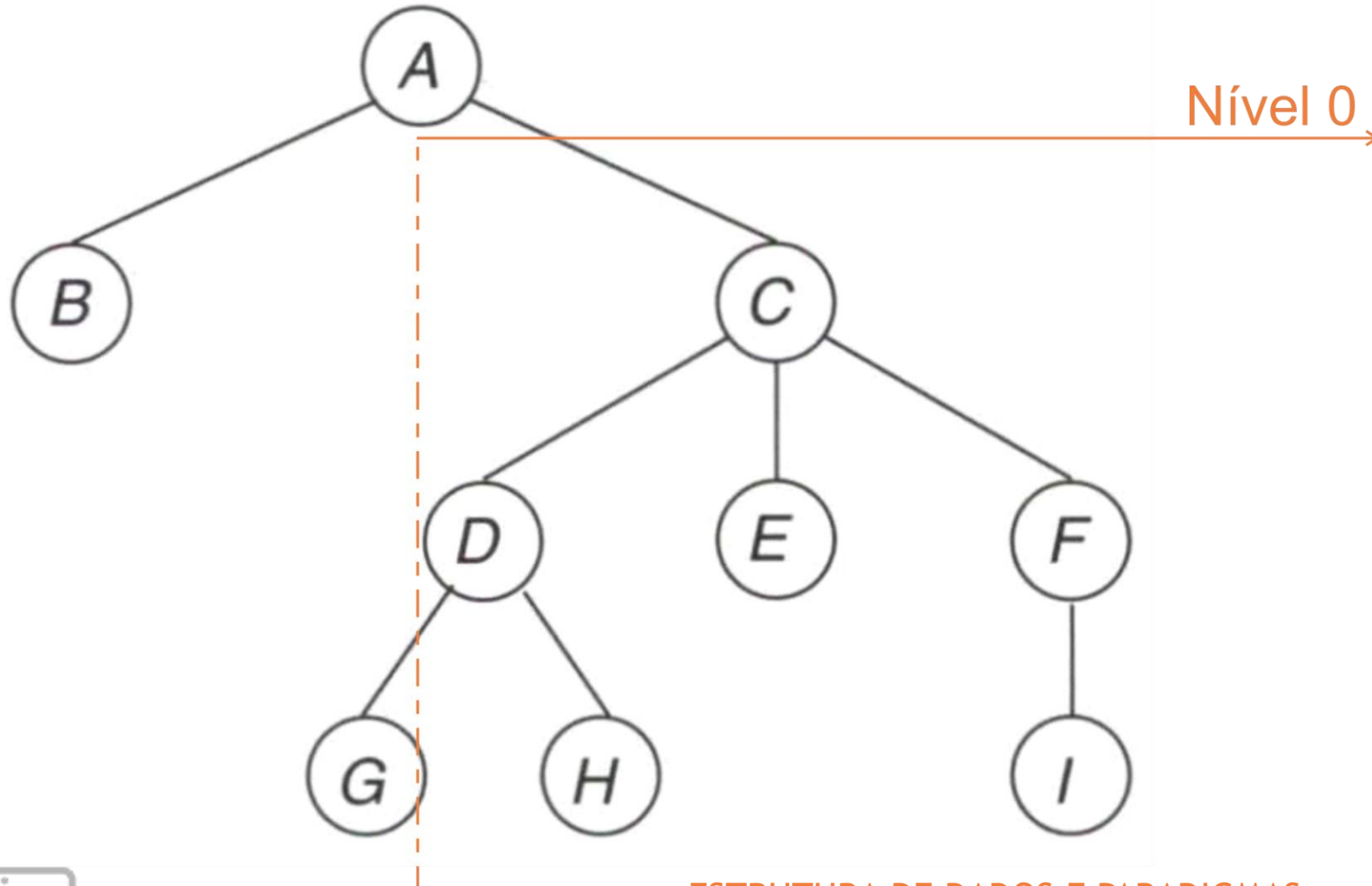


Elementos de uma Árvore

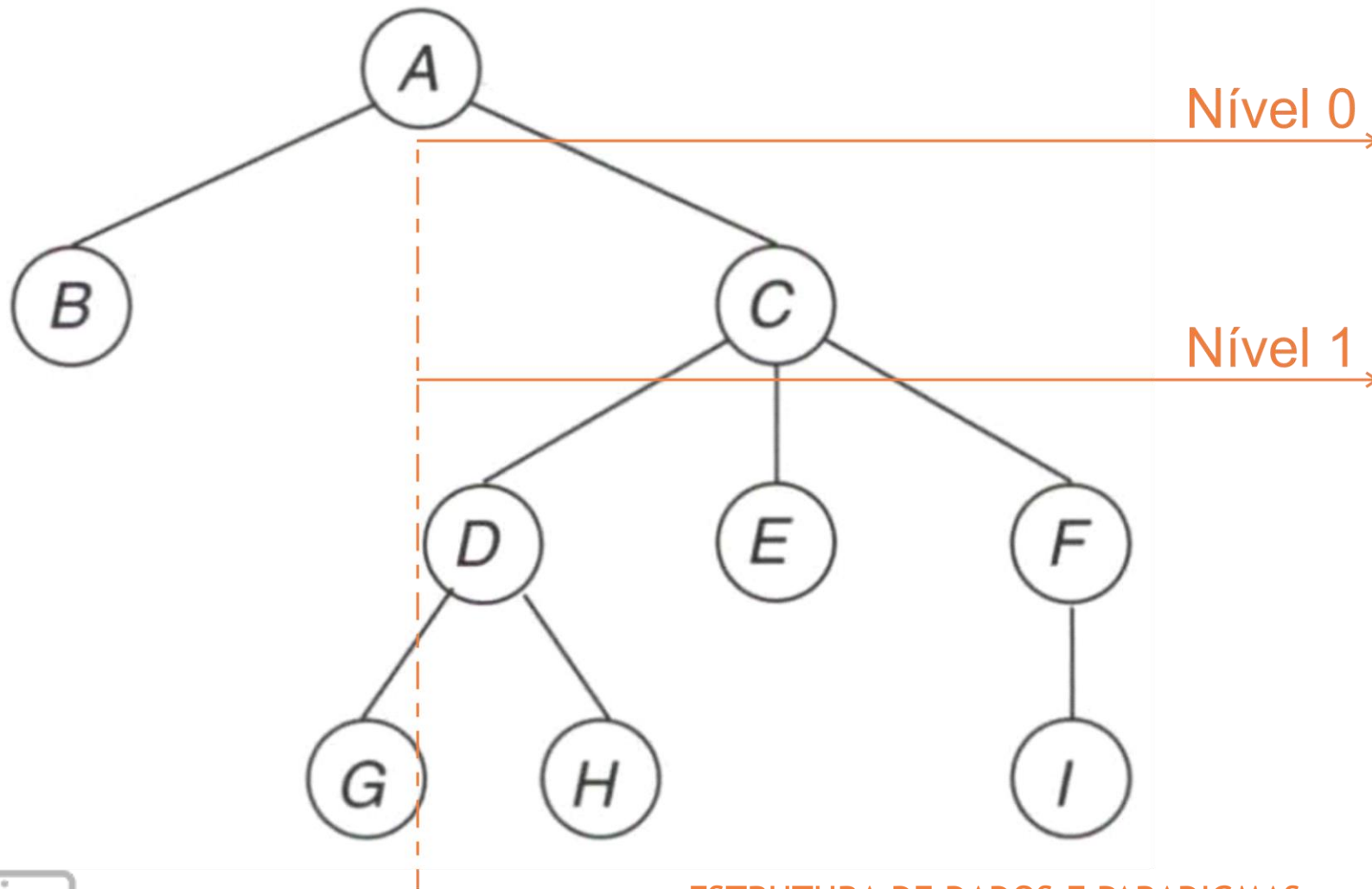


ESTRUTURA DE DADOS E PARADIGMAS
REVISÃO DE LISTAS LINEARES, PILHAS, FILAS E BUSCA BINÁRIA E INTRODUÇÃO À
ÁRVORES

Elementos de uma Árvore



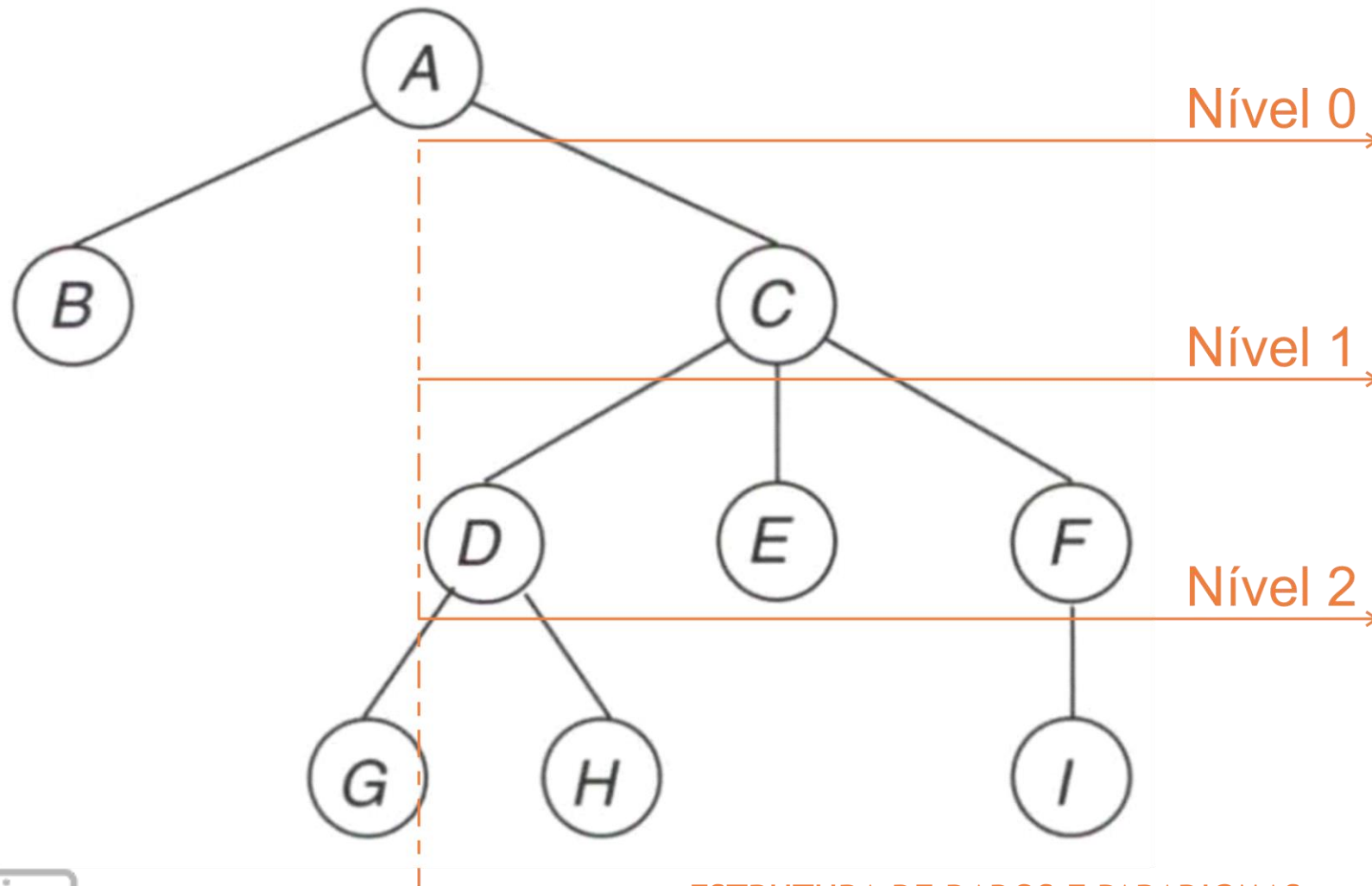
Elementos de uma Árvore



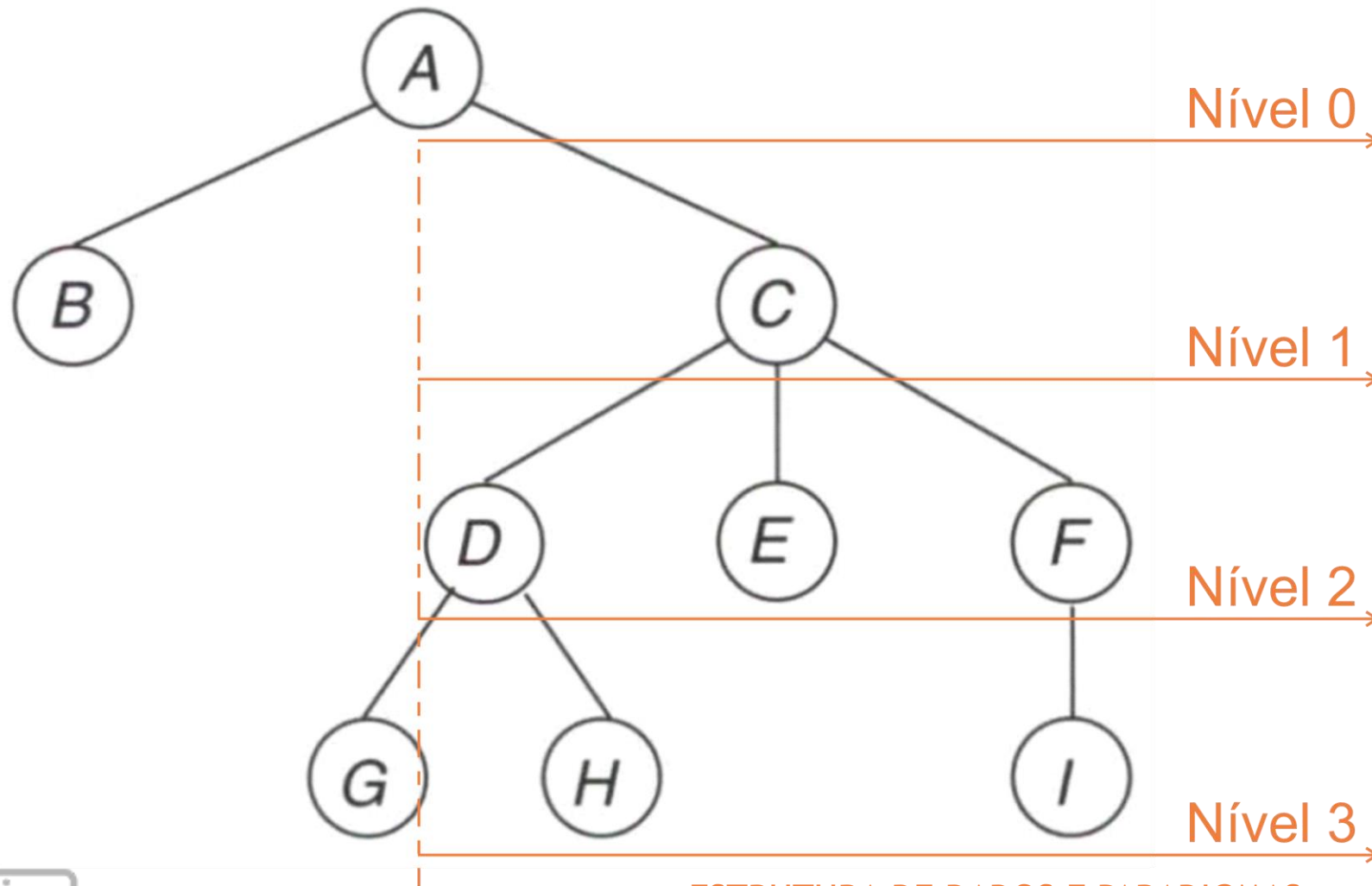
ESTRUTURA DE DADOS E PARADIGMAS
REVISÃO DE LISTAS LINEARES, PILHAS, FILAS E BUSCA BINÁRIA E INTRODUÇÃO À
ÁRVORES



Elementos de uma Árvore



Elementos de uma Árvore



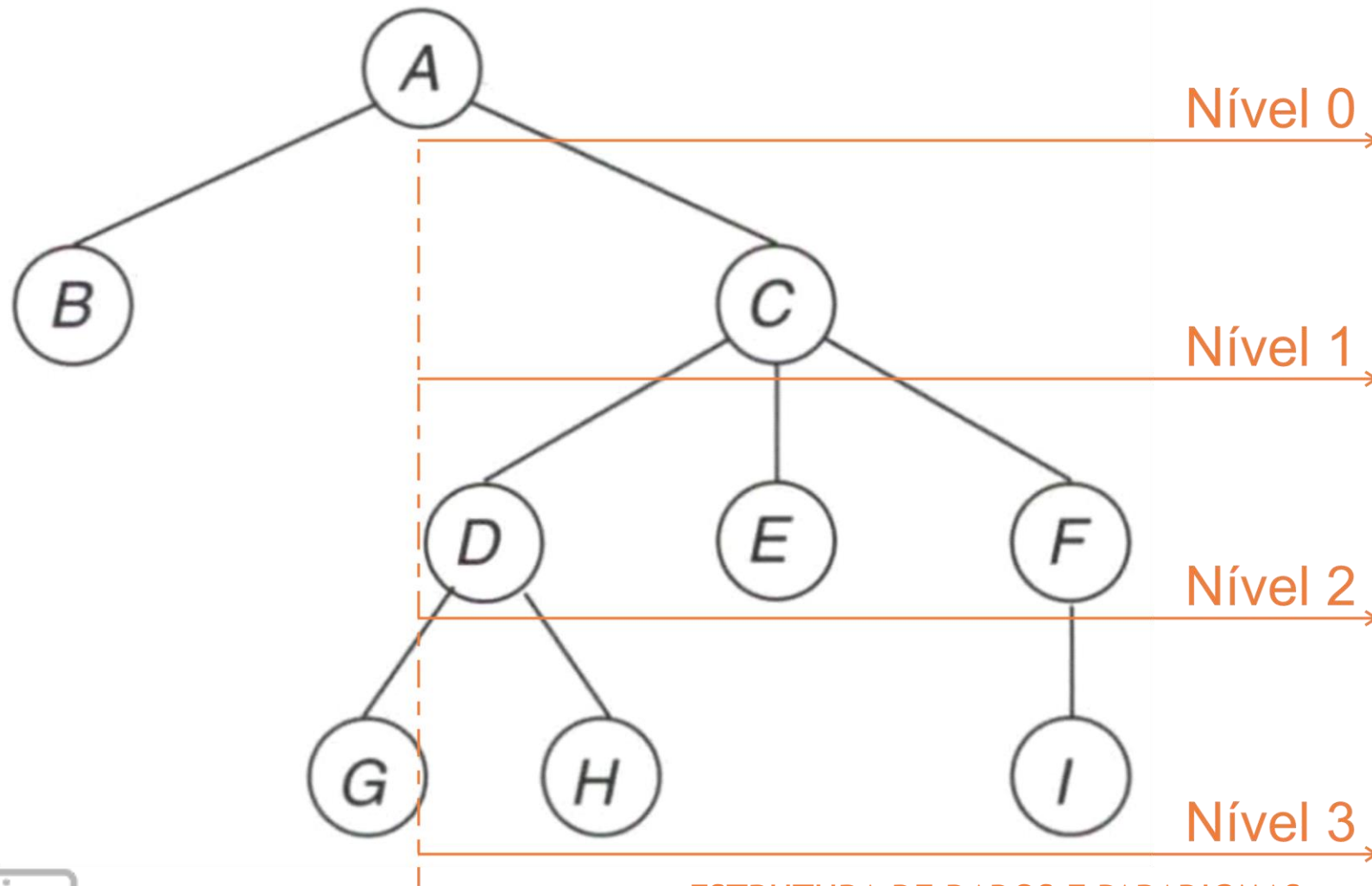
Elementos de uma Árvore

- **Altura:**

- É o **nível** do **nó folha** que tem o mais **longo caminho** até a **raiz**, somando **1**.



Elementos de uma Árvore



$$h(T) = 3 + 1 = 4$$

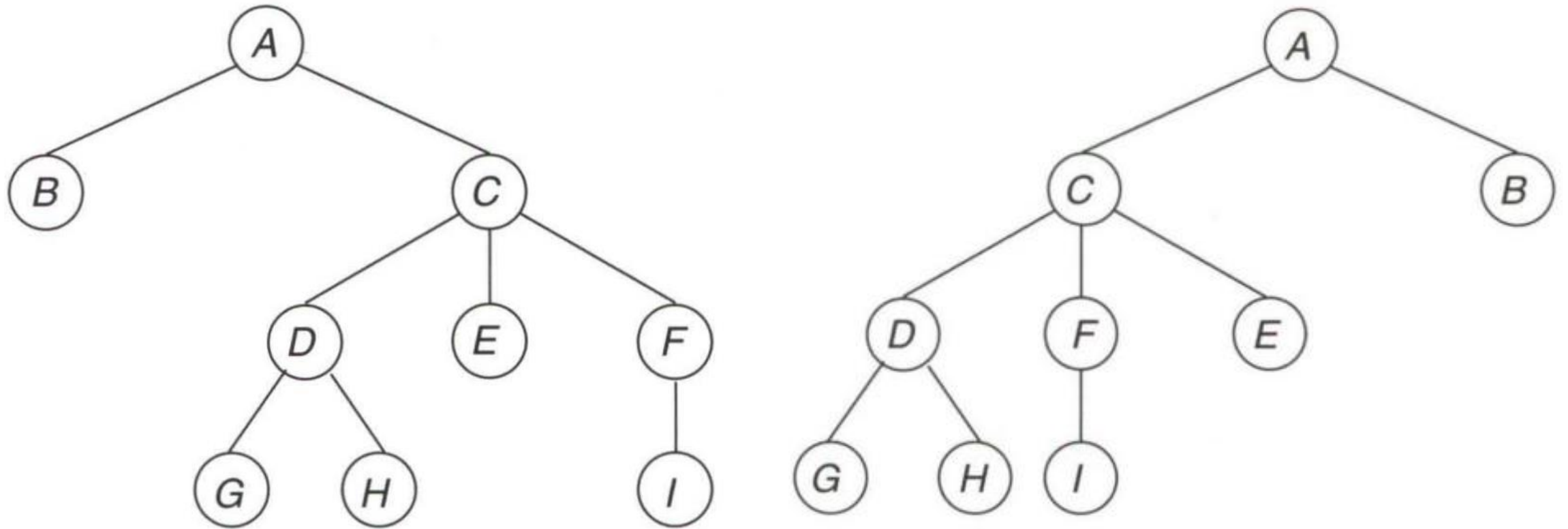


Elementos de uma Árvore

- **Árvore ordenada:**
 - Os filhos de cada nó estão **ordenados** (assume-se ordenação da esquerda para a direita).



Elementos de uma Árvore

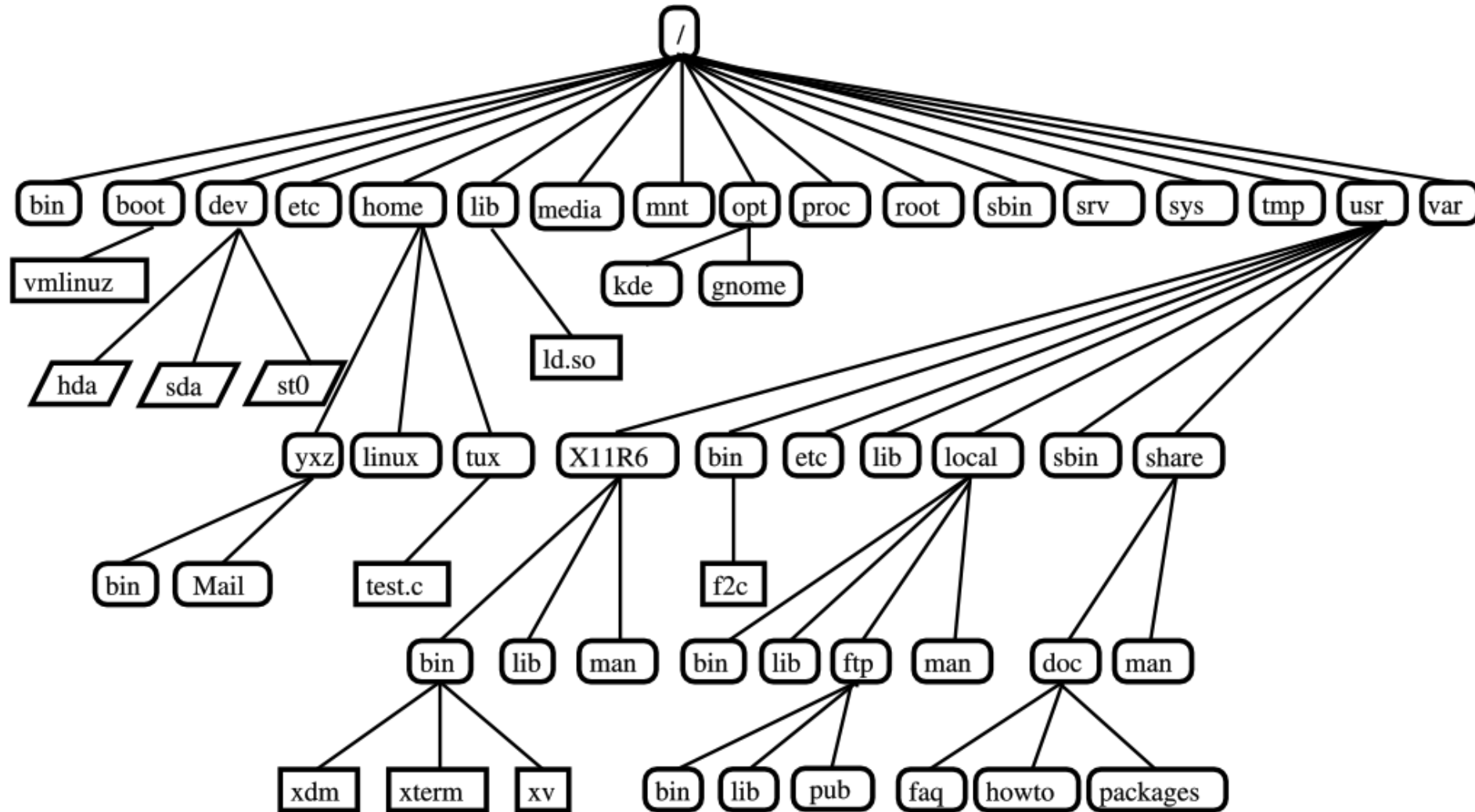


Elementos de uma Árvore

- A definição de árvore **não impõe** qualquer condição sobre o **número** de **filhos** de um **nó**:
 - Pode variar de **0** a **qualquer inteiro**.



Elementos de uma Árvore



ESTRUTURA DE DADOS E PARADIGMAS
REVISÃO DE LISTAS LINEARES, PILHAS, FILAS E BUSCA BINÁRIA E INTRODUÇÃO À
ÁRVORES

Árvores Binárias

- Uma árvore T é um conjunto finito de elementos denominados **nós**, tal que:
 - $T = 0$, árvore vazia, ou



Árvores Binárias

- $T > 0$
 - Existe um nó especial r denominado **raiz** de T .
 - Os nós restantes (**nós filhos**) podem ser divididos em **dois** sub-conjuntos disjuntos, TrE e TrD , a subárvore esquerda e a direita de r , respectivamente, as quais são também árvores binárias.

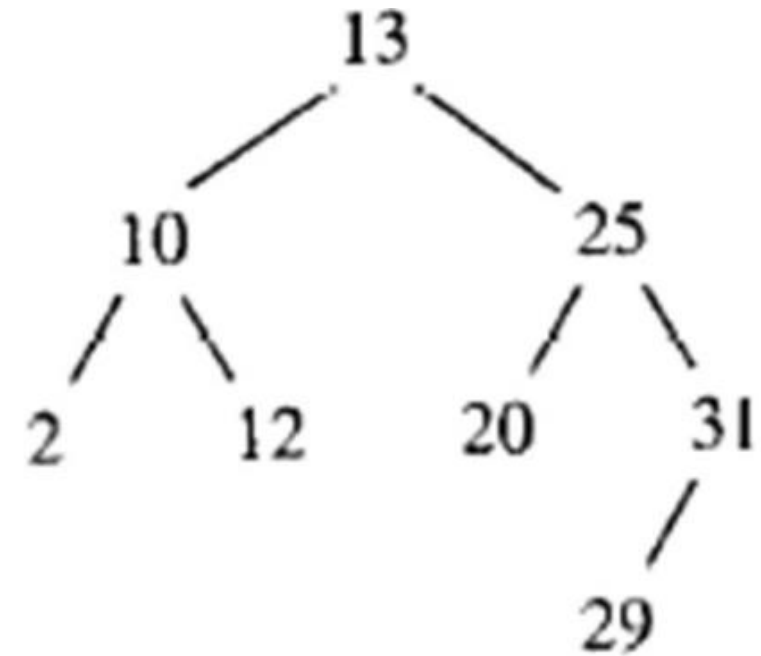
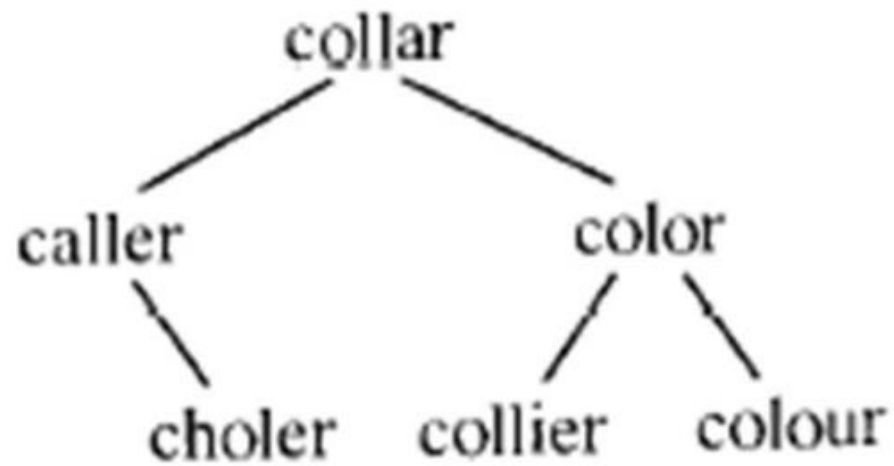


Árvores Binárias

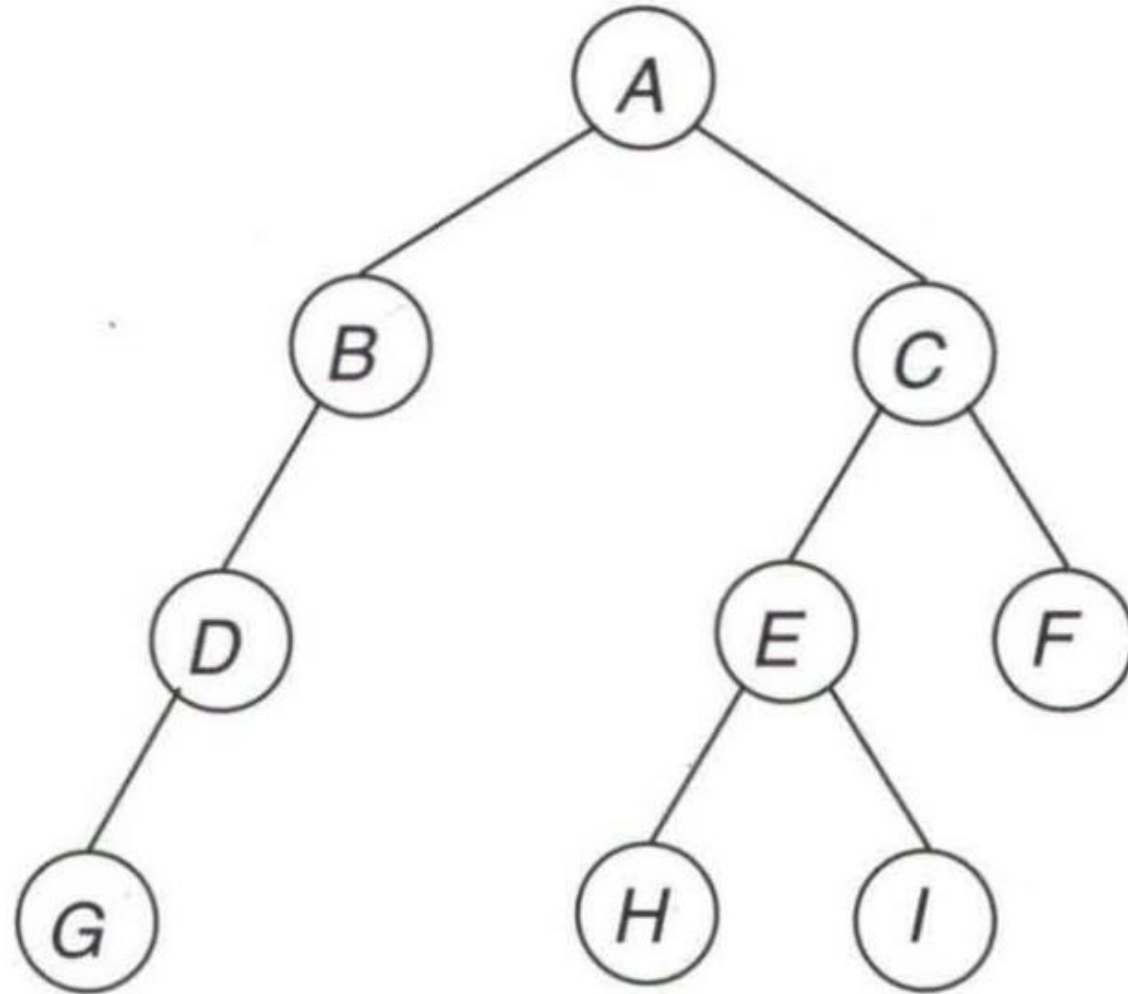
- Portanto, a **Árvore Binária** tem grau máximo **2**.
- Nó filho **ESQUERDO** e nó filho **DIREITO**.



Árvores Binárias



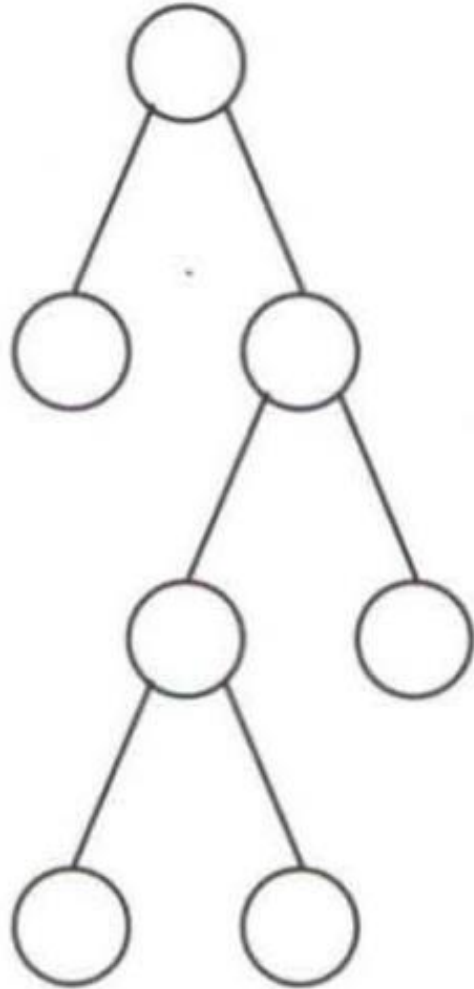
Árvores Binárias



ESTRUTURA DE DADOS E PARADIGMAS
REVISÃO DE LISTAS LINEARES, PILHAS, FILAS E BUSCA BINÁRIA E INTRODUÇÃO À
ÁRVORES



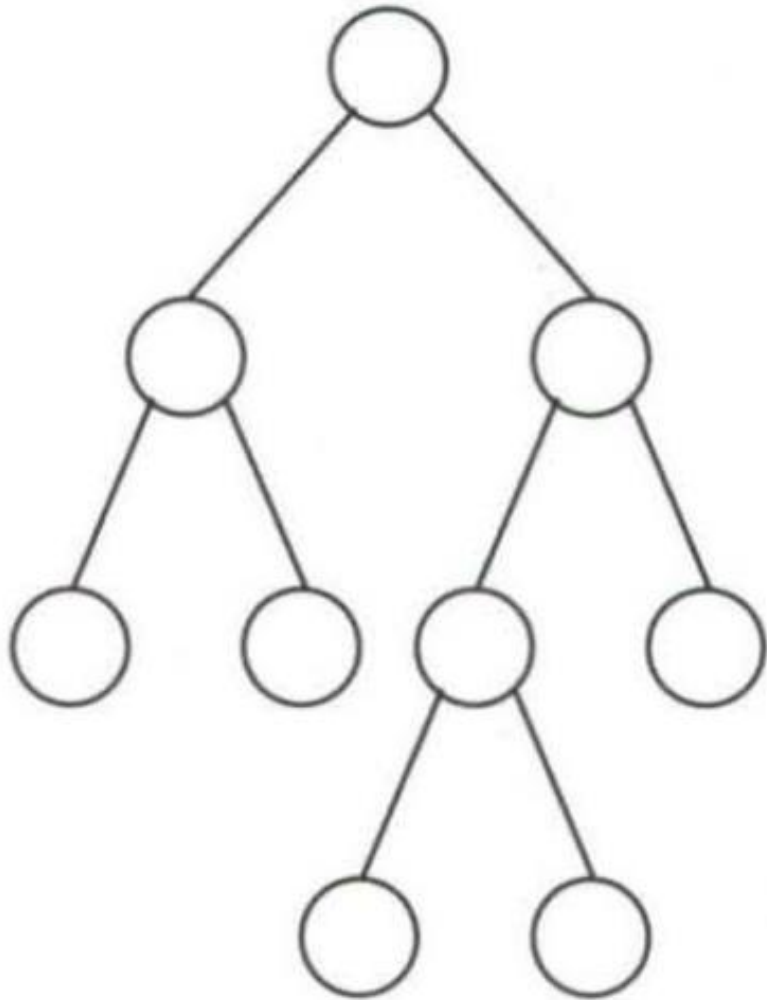
Árvores Binárias



- **Árvore estritamente binária:** cada nó possui 0 ou 2 filhos.



Árvores Binárias

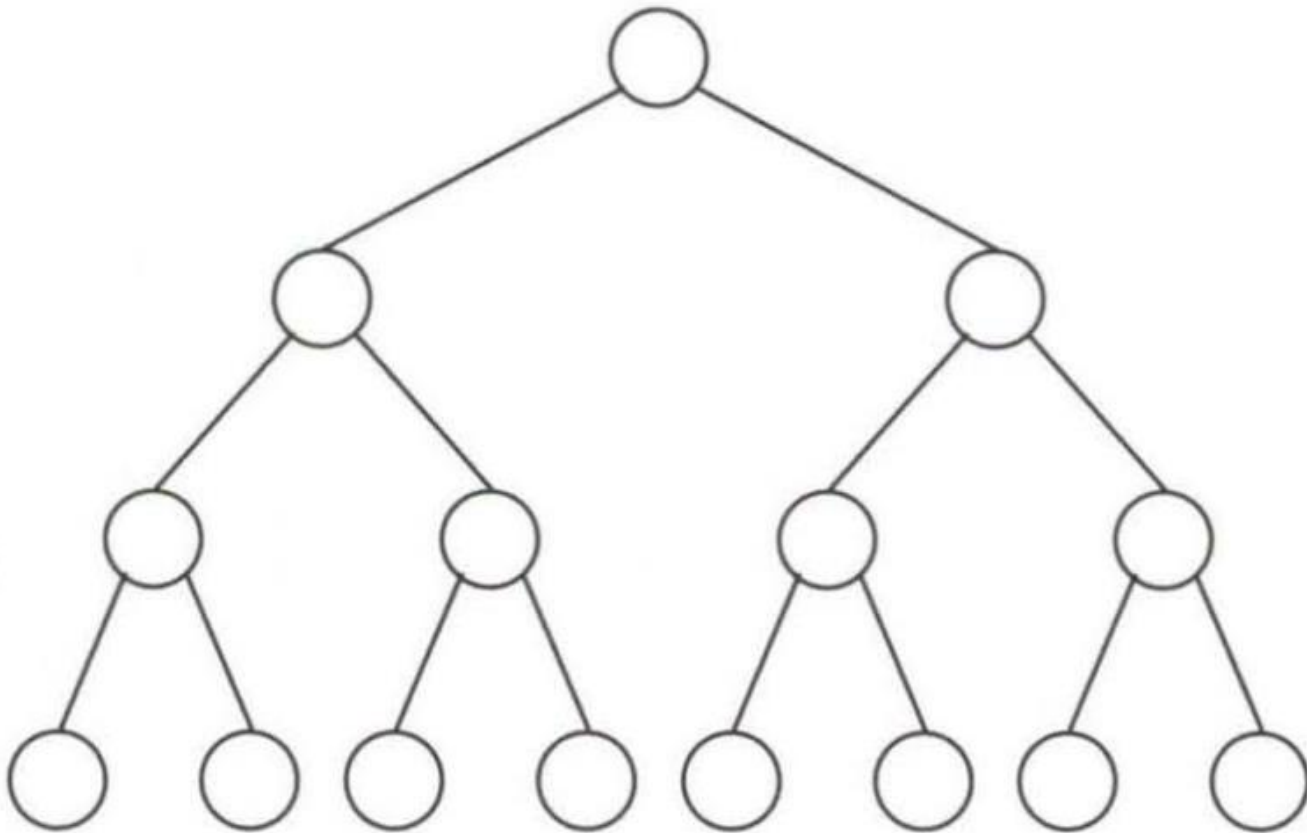


- **Árvore binária completa:** se v é um nó tal que alguma subárvore de v é vazia, então v se localiza ou no último (maior) ou no penúltimo nível da árvore.

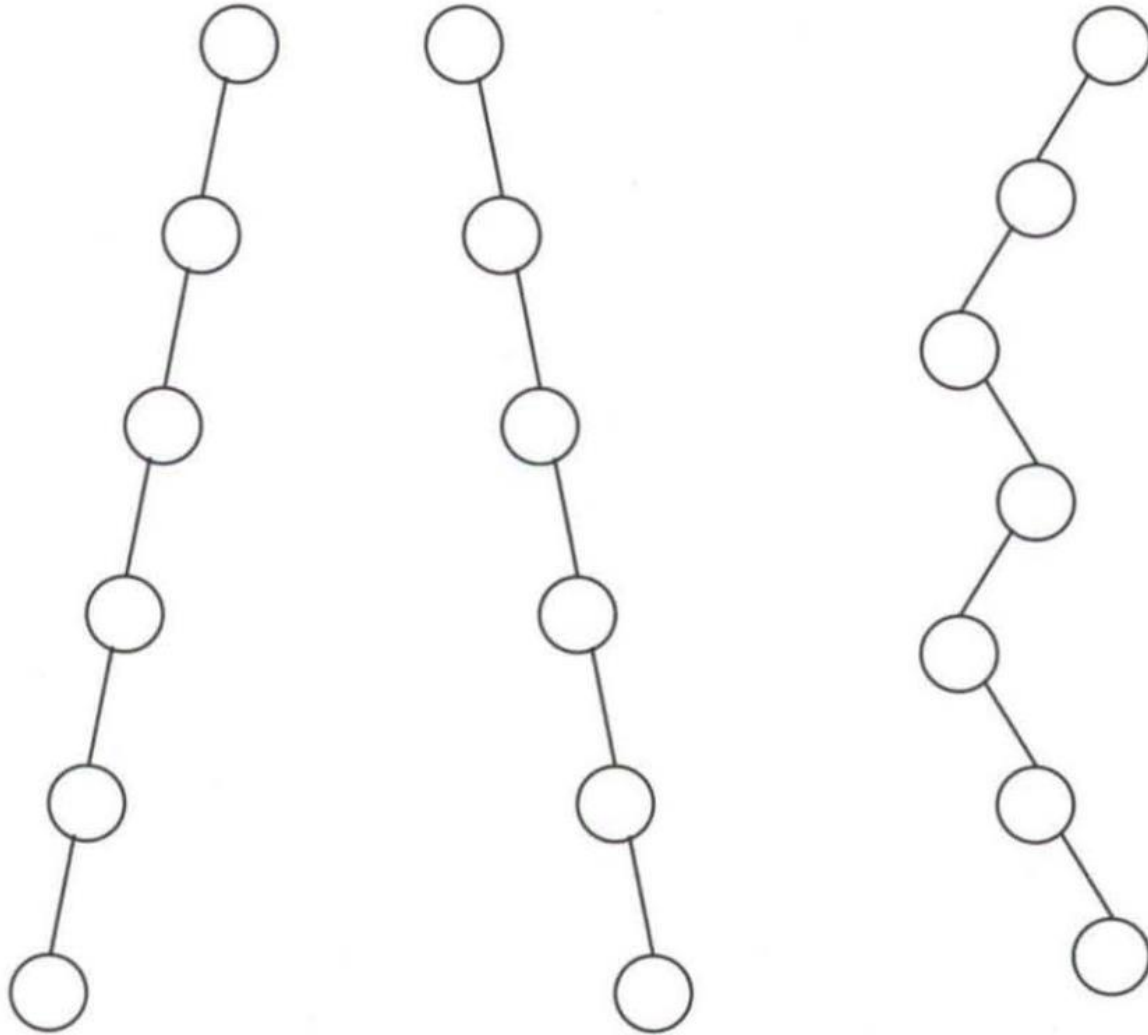


Árvores Binárias

- **Árvore binária cheia:** Se v é um nó tal que alguma subárvore de v é vazia, então v se localiza no último (maior) nível da árvore. v é um nó folha.



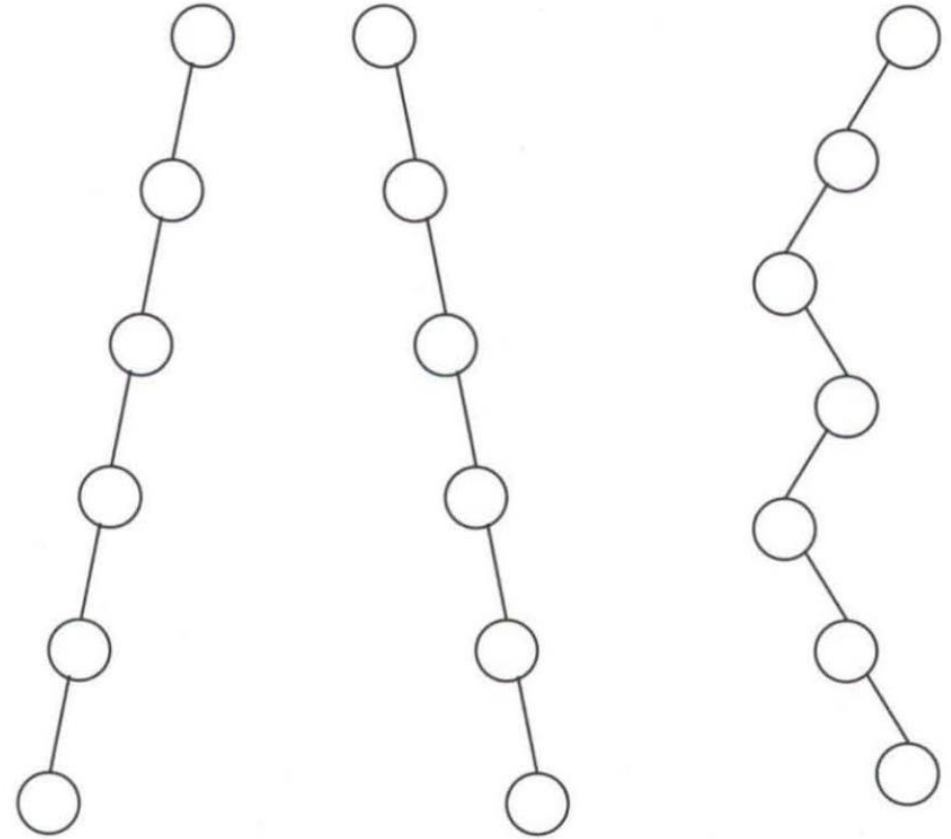
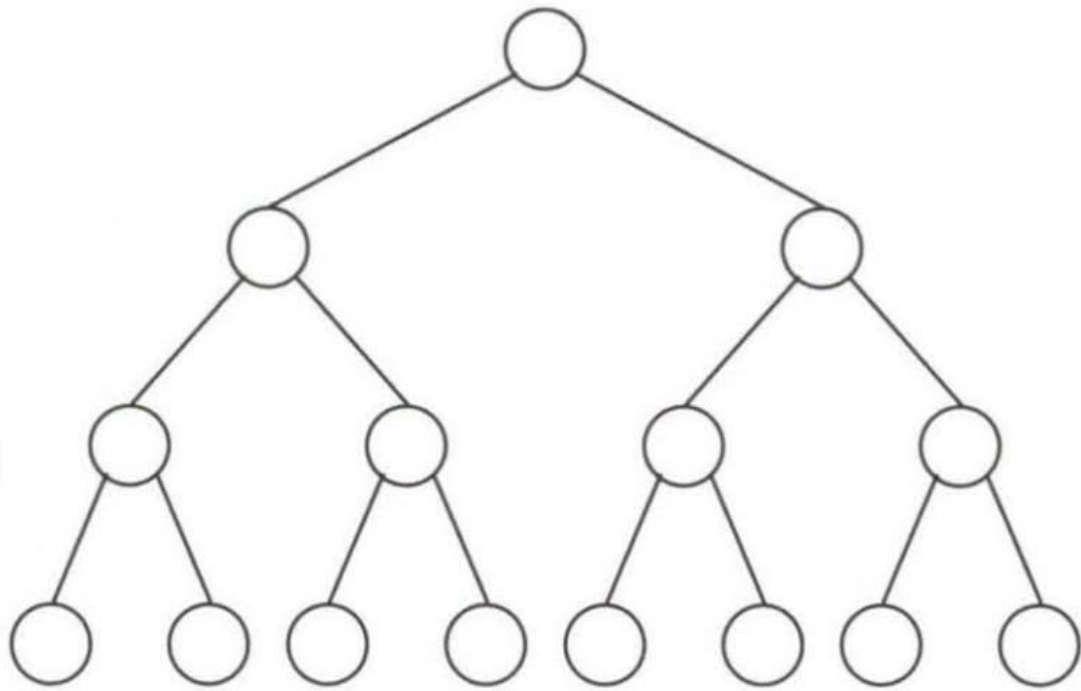
Árvores Binárias



- **Árvore zigzague:** árvore binária que possui altura máxima, ou seja, é igual a **n** nós.



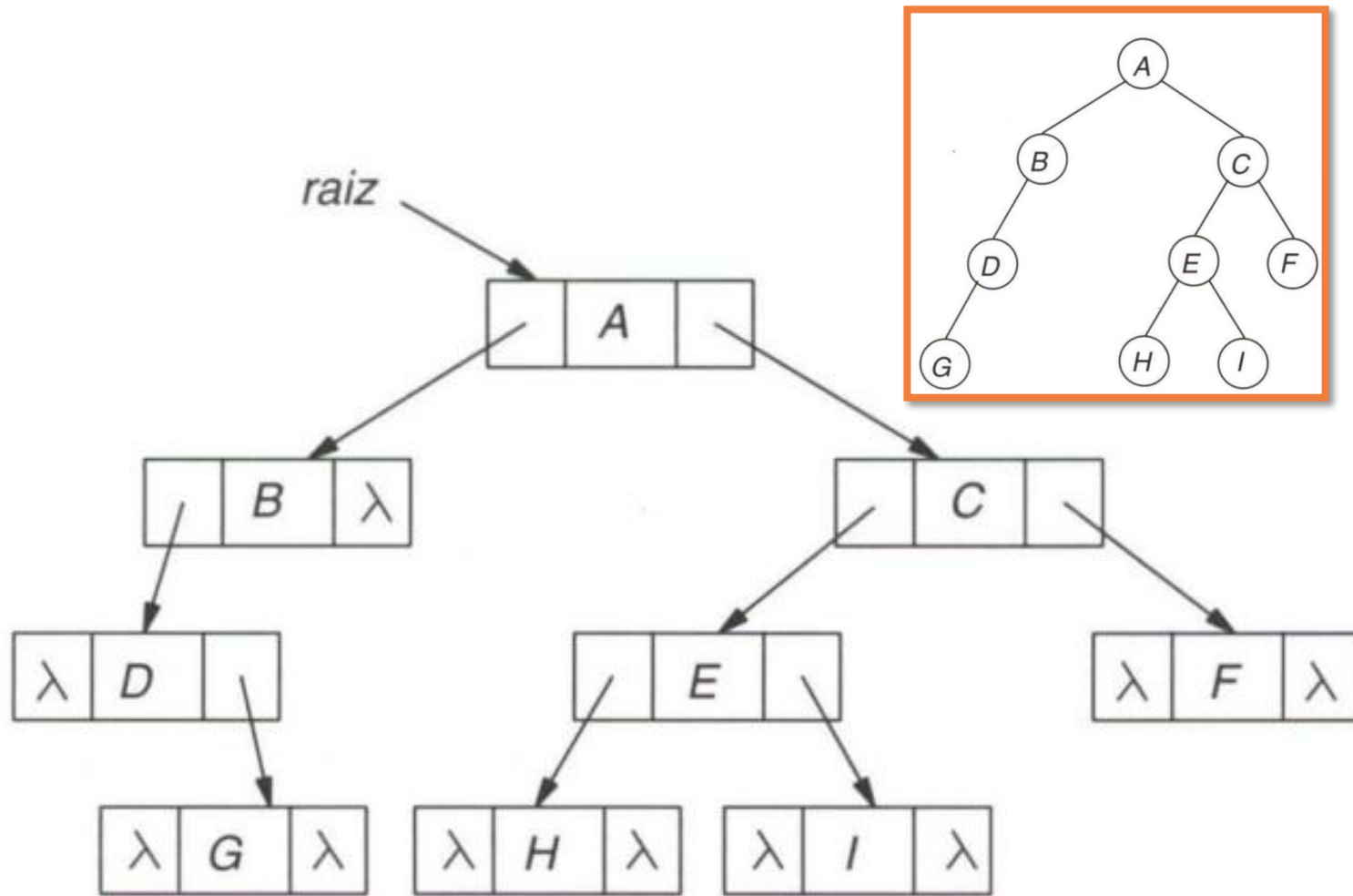
Árvores Binárias



Qual é mais eficiente?



Árvores Binárias



- Representação encadeada de árvores binária
- Campo de informação.
- **Ponteiro** para nó esquerdo.
- **Ponteiro** para nó direito.



Percurso em Árvores Binárias

- Visita **sistemática** a **cada nó** da Árvore.
- Operação básica para **manipulação** de árvores.
- Percorrer a Árvore **significa** visitar **todos** os seus **nós** uma vez.
- Para ter um **algoritmo de percurso**, resta **definir** a **ordem** das **operações**:

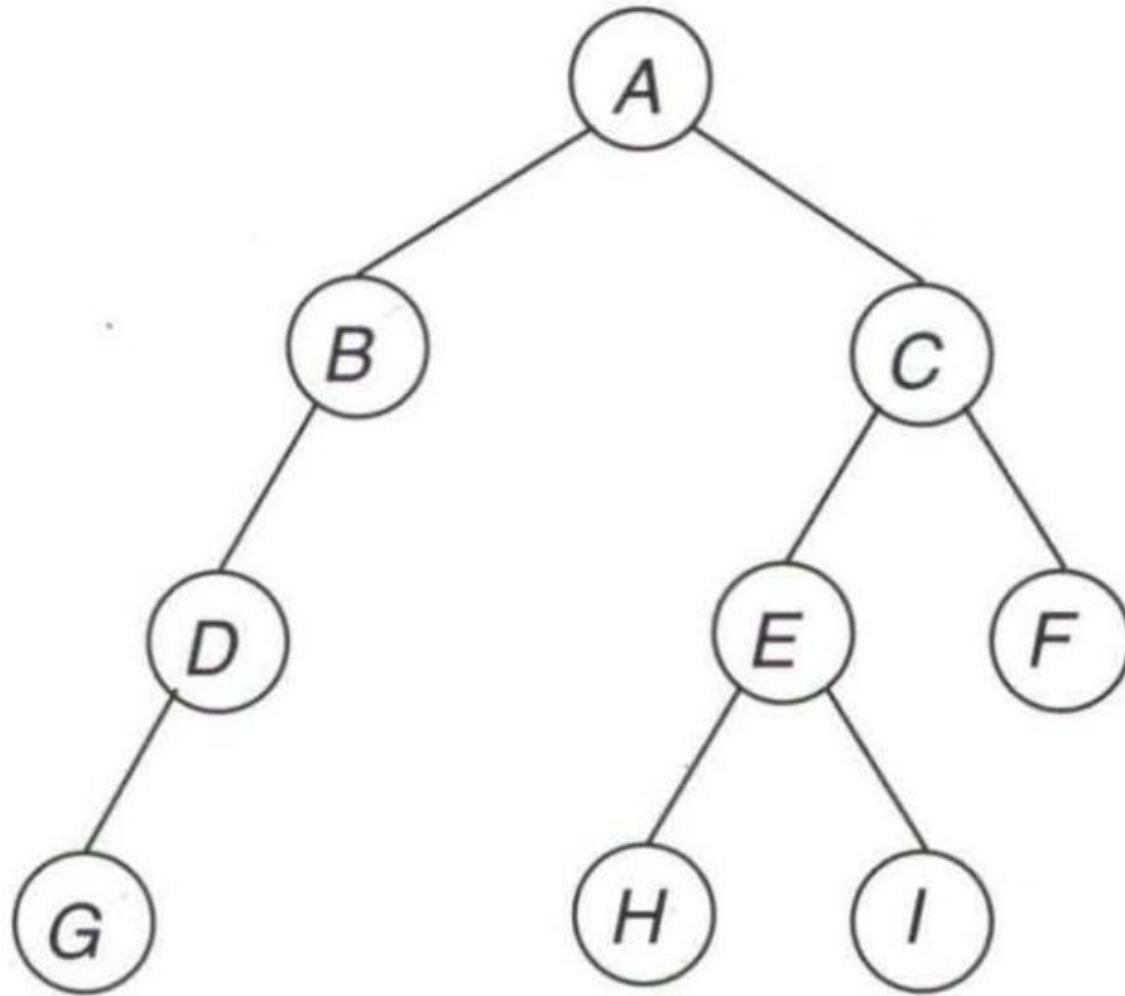


Percurso em Árvores Binárias

- Percurso em **pré-ordem**:
 - Visitar a raiz;
 - Percorrer sua sub árvore esquerda, em pré-ordem;
 - Percorrer sua sub árvore direita, em pré-ordem.



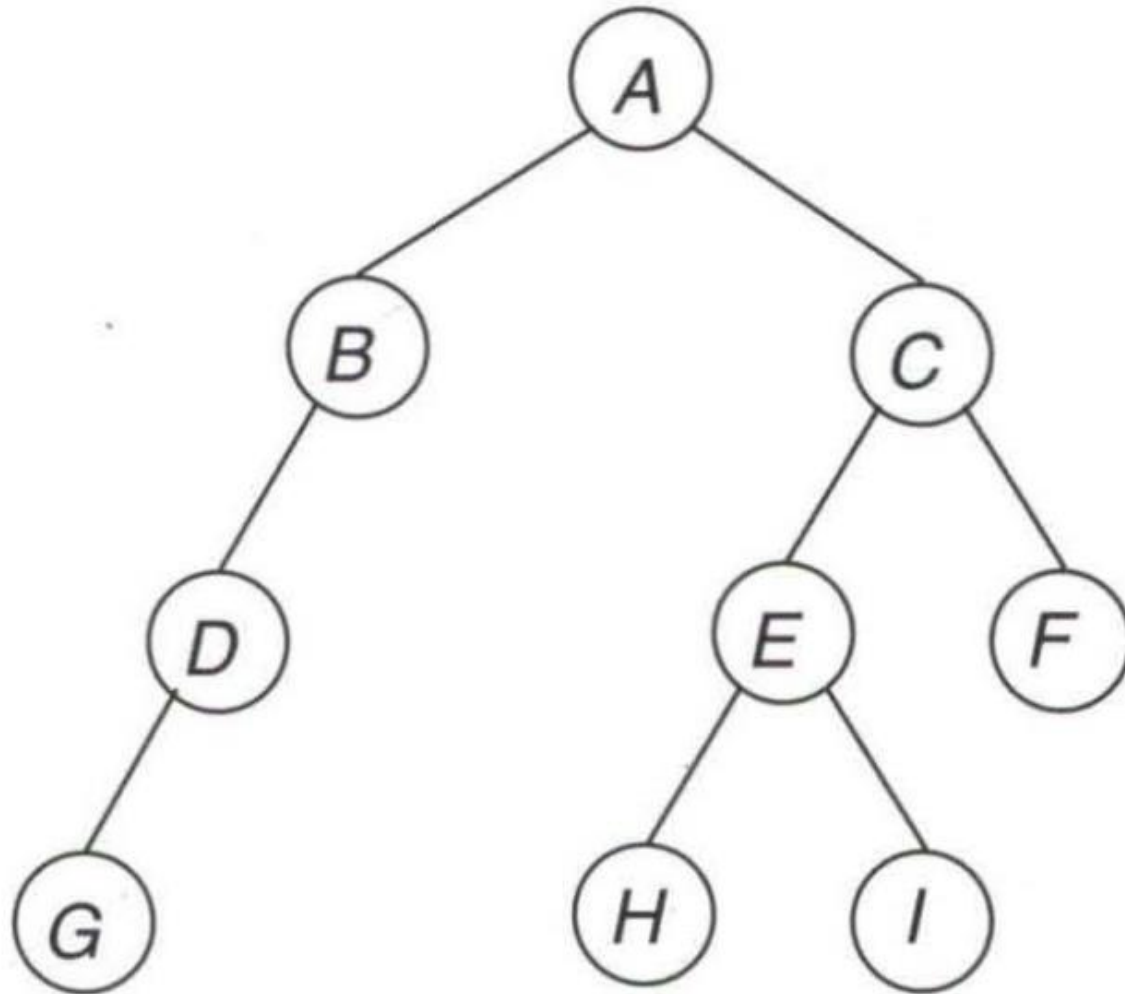
Percurso em Árvores Binárias



- **Percurso em pré-ordem:**
- ???



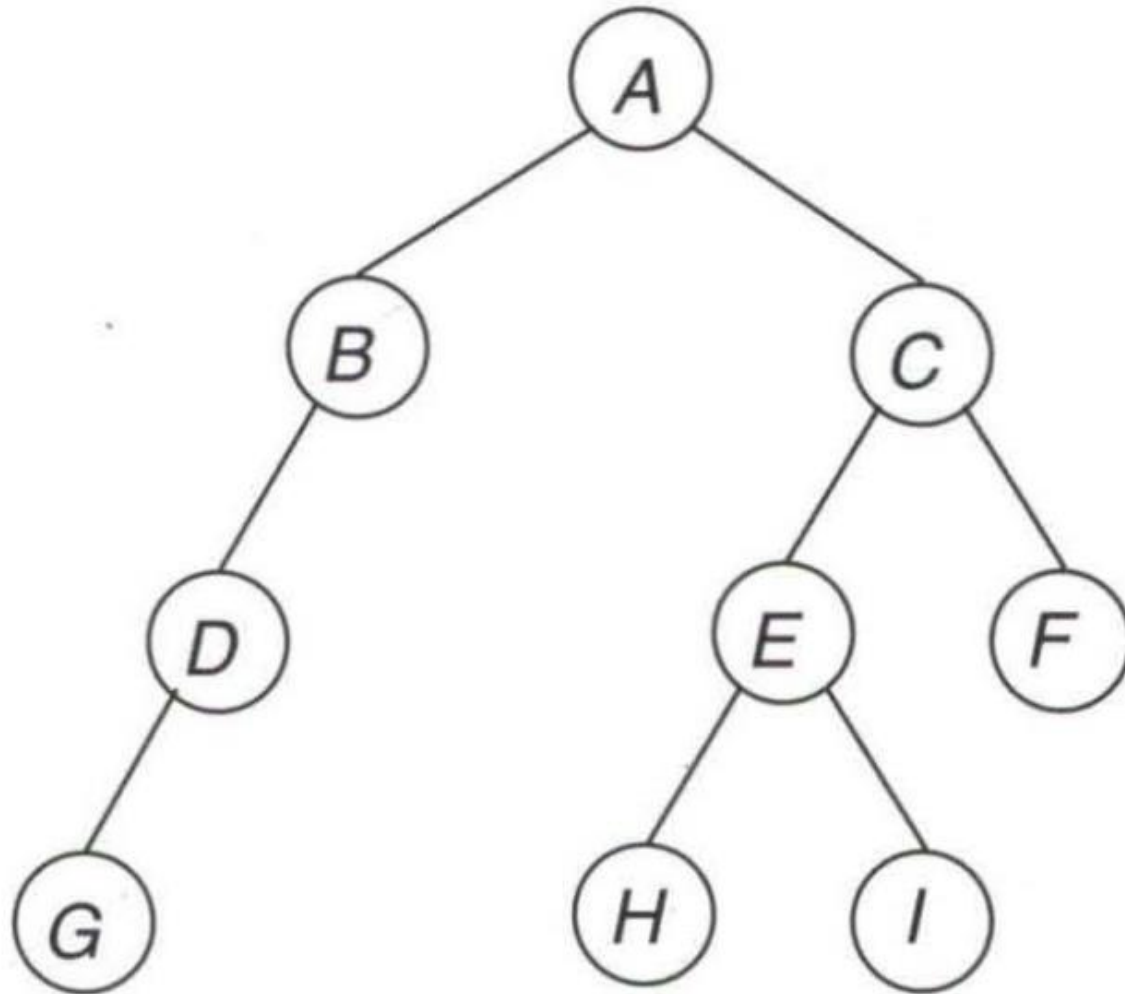
Percurso em Árvores Binárias



- Percurso em pré-ordem:
- A



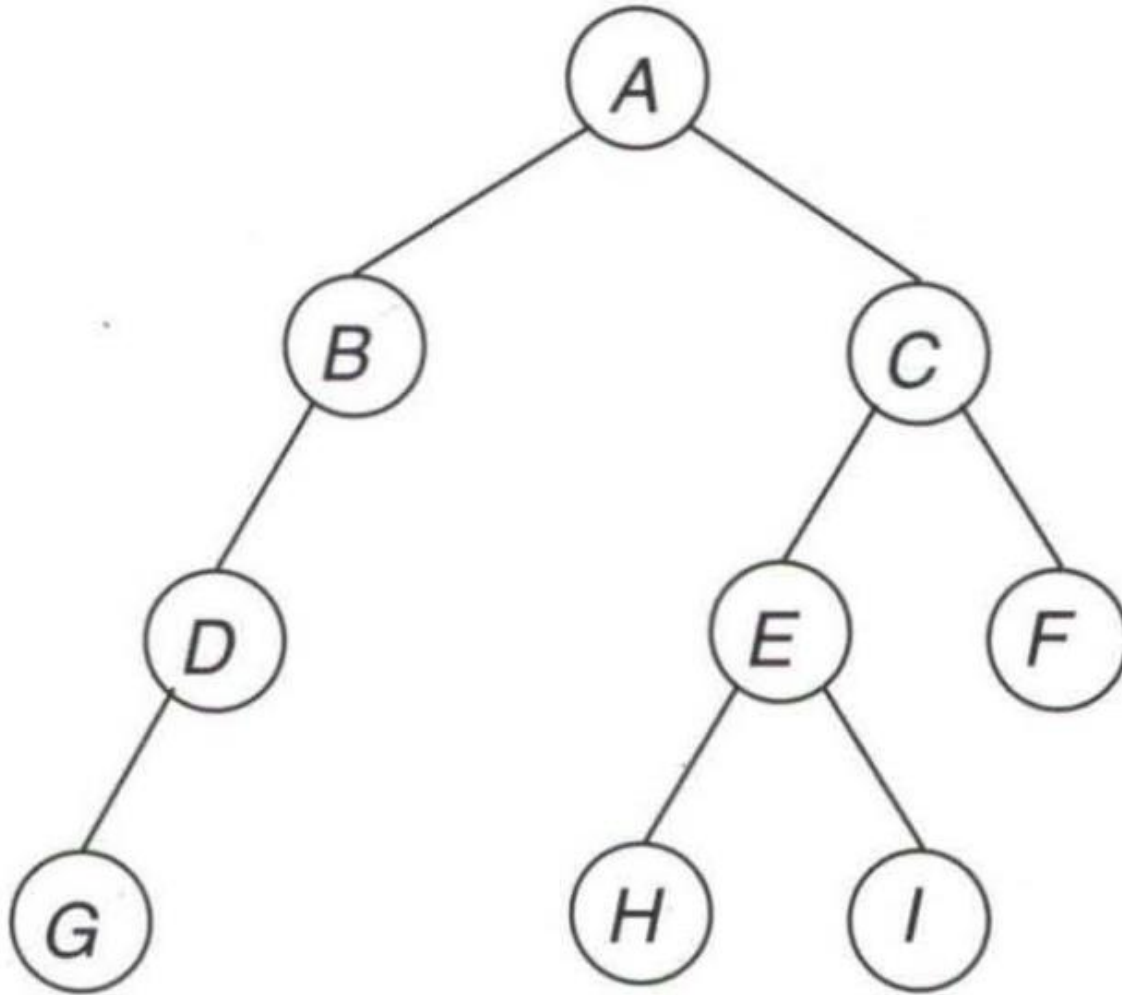
Percurso em Árvore Binárias



- **Percurso em pré-ordem:**
- **A B**



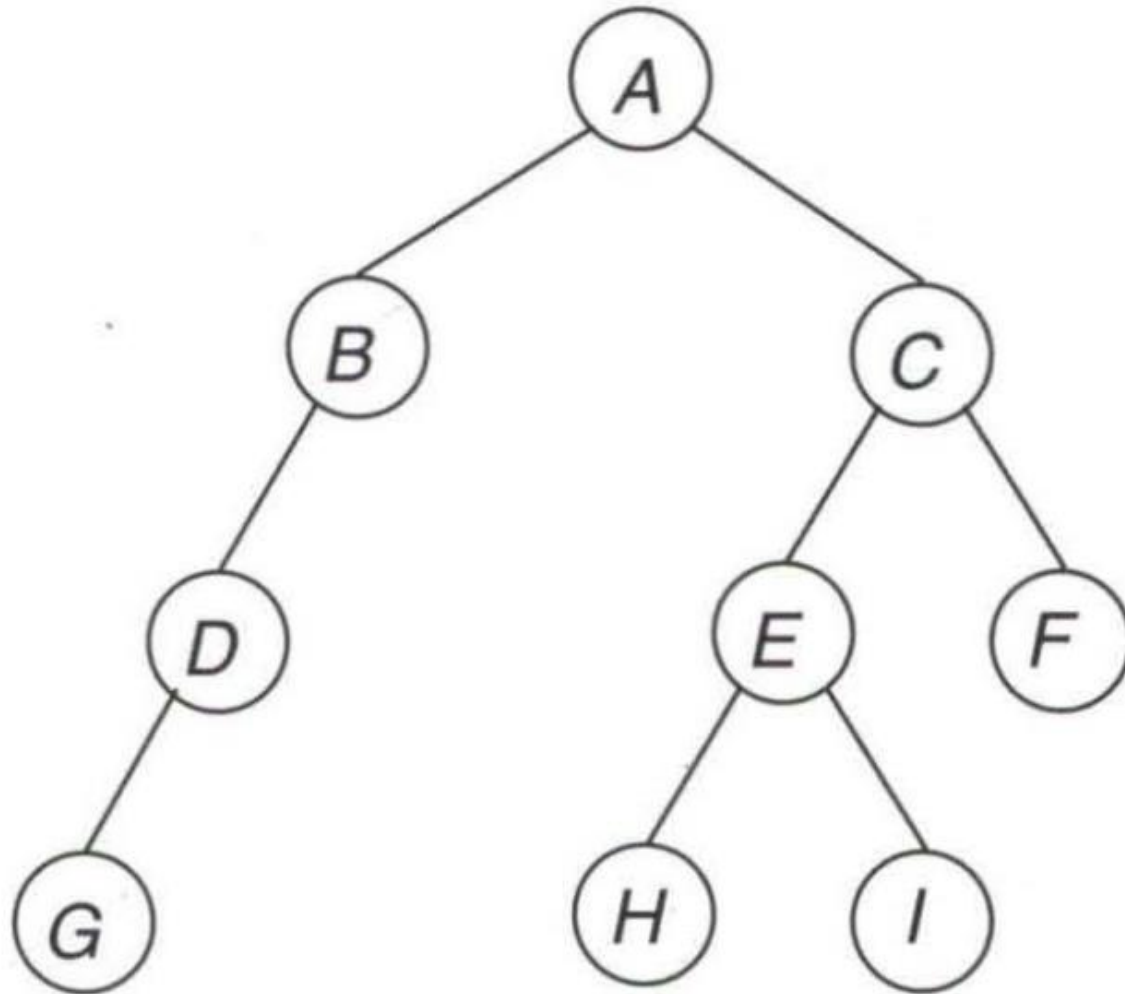
Percurso em Árvore Binárias



- **Percurso em pré-ordem:**
- **A B D**



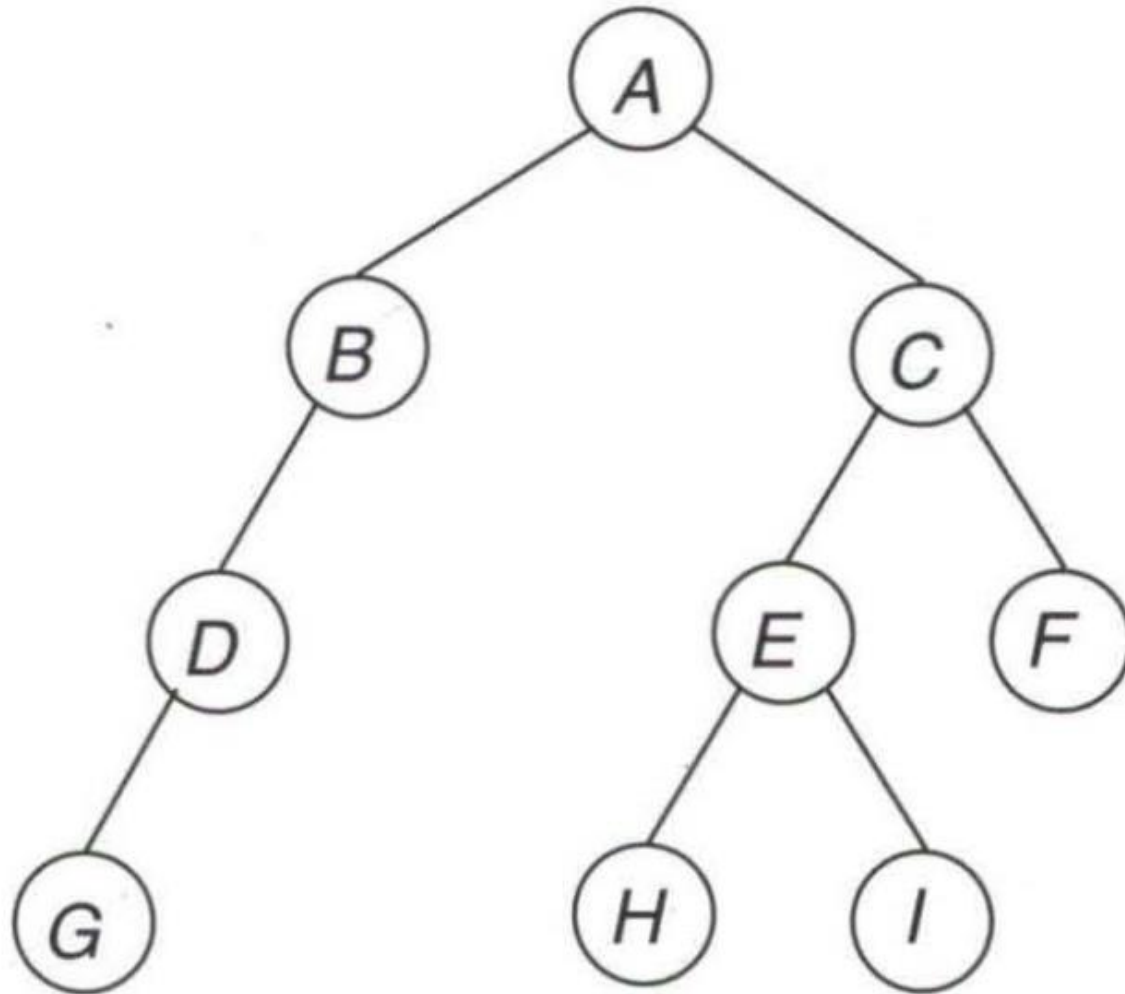
Percurso em Árvores Binárias



- **Percurso em pré-ordem:**
- **A B D G**



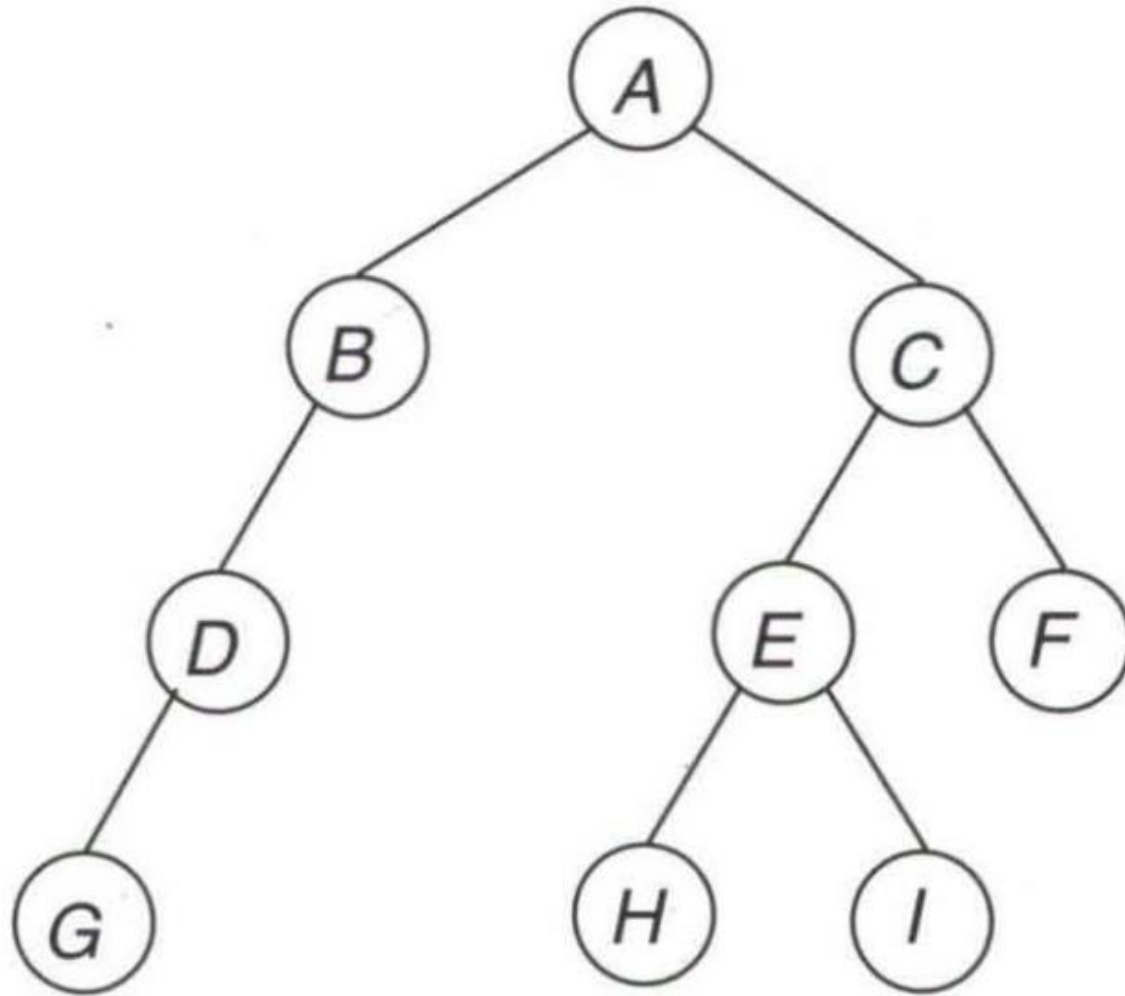
Percurso em Árvores Binárias



- **Percurso em pré-ordem:**
- **A B D G C**



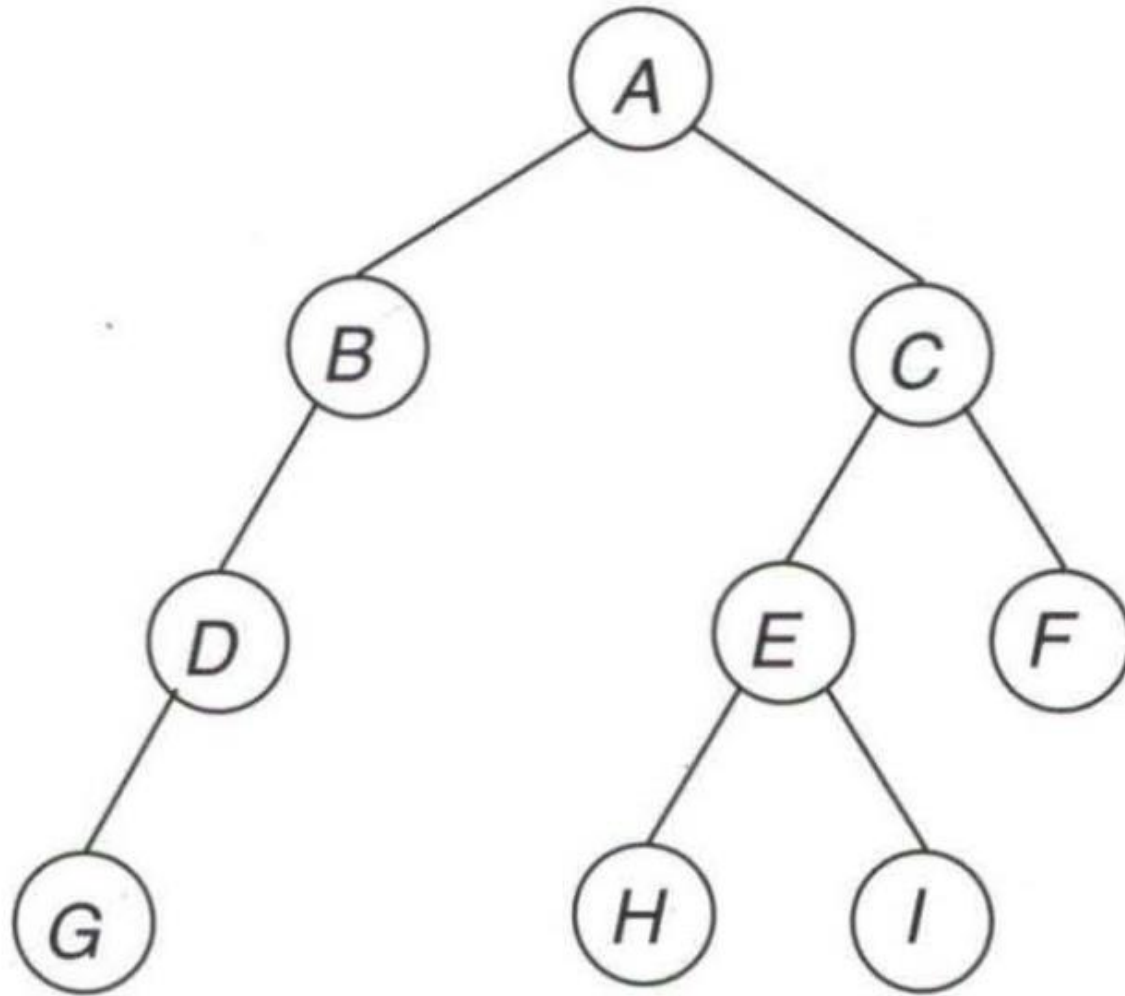
Percurso em Árvores Binárias



- Percurso em pré-ordem:
- **A B D G C E**



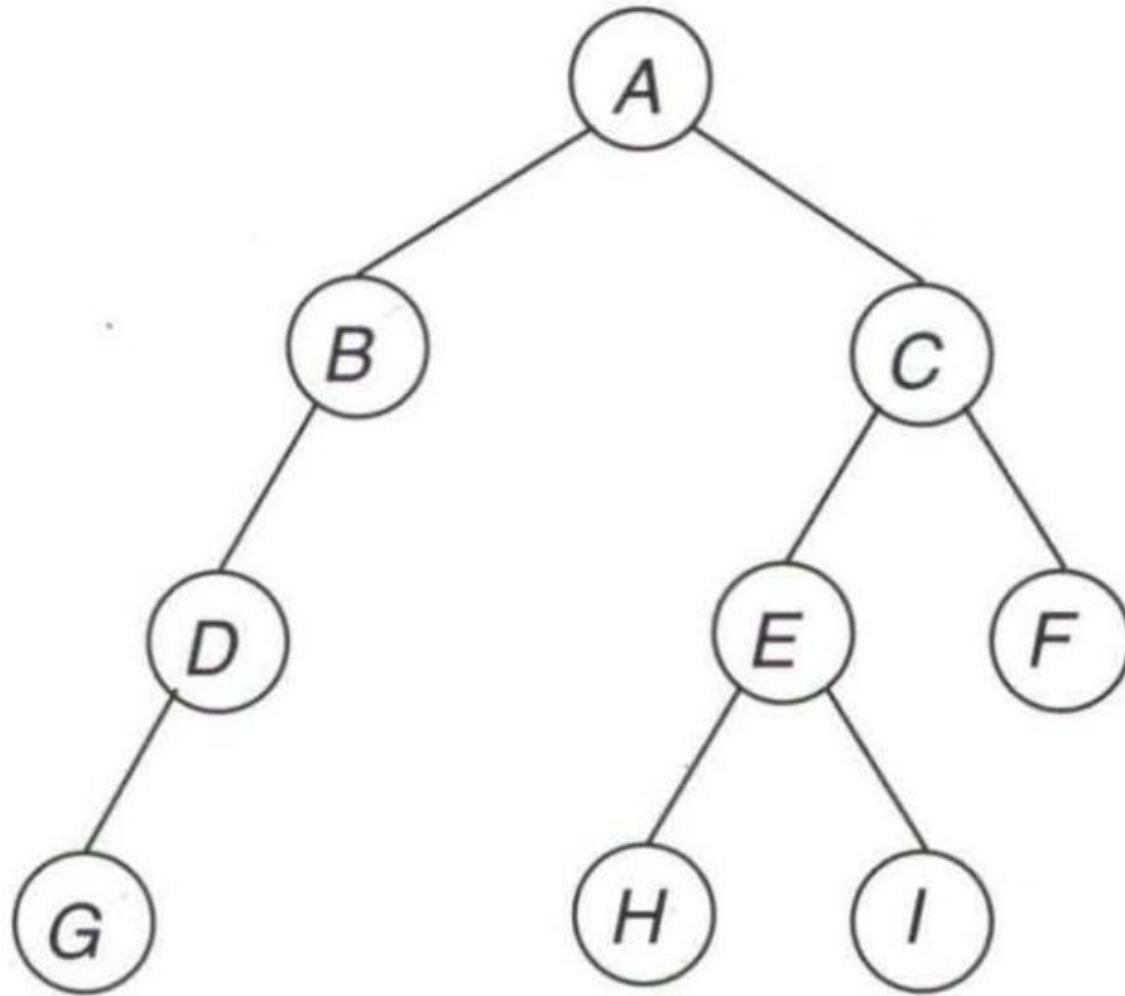
Percurso em Árvores Binárias



- Percurso em pré-ordem:
- **A B D G C E H**



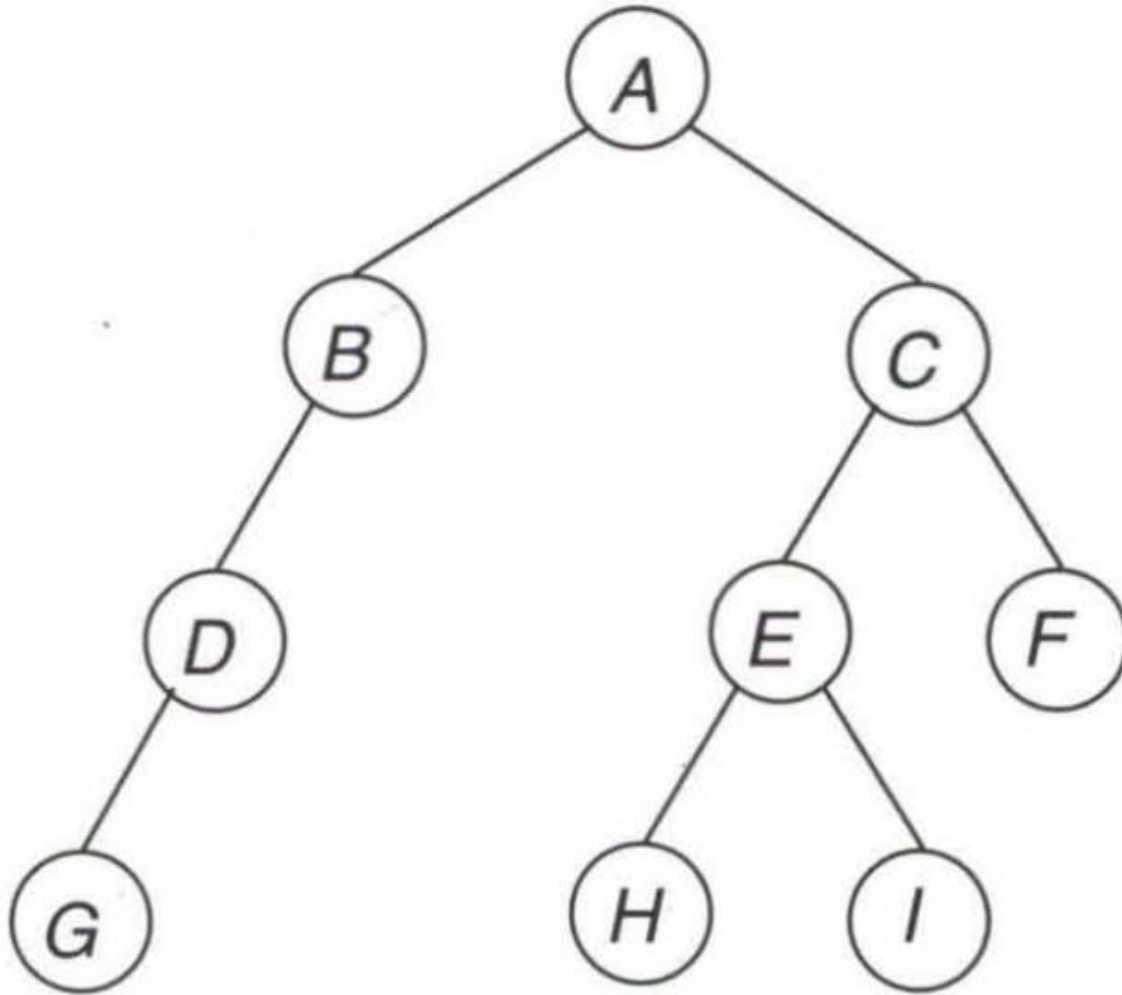
Percurso em Árvores Binárias



- Percurso em pré-ordem:
- **A B D G C E H I**



Percurso em Árvores Binárias



- Percurso em pré-ordem:
- **A B D G C E H I F**

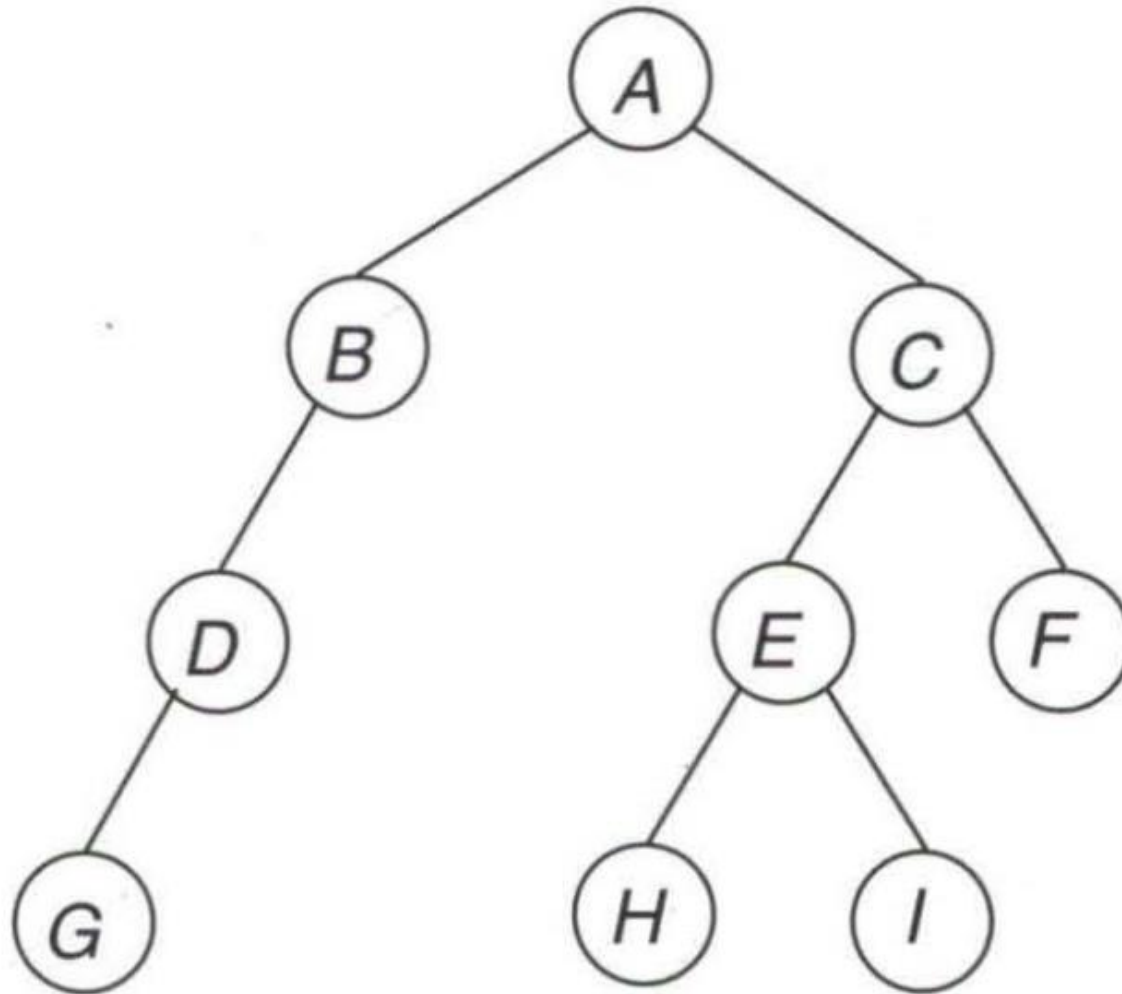


Percurso em Árvores Binárias

- Percurso em **ordem simétrica**:
 - Percorrer sua sub árvore esquerda, em **ordem simétrica**;
 - Visitar a raiz;
 - Percorrer sua sub árvore direita, **ordem simétrica**.



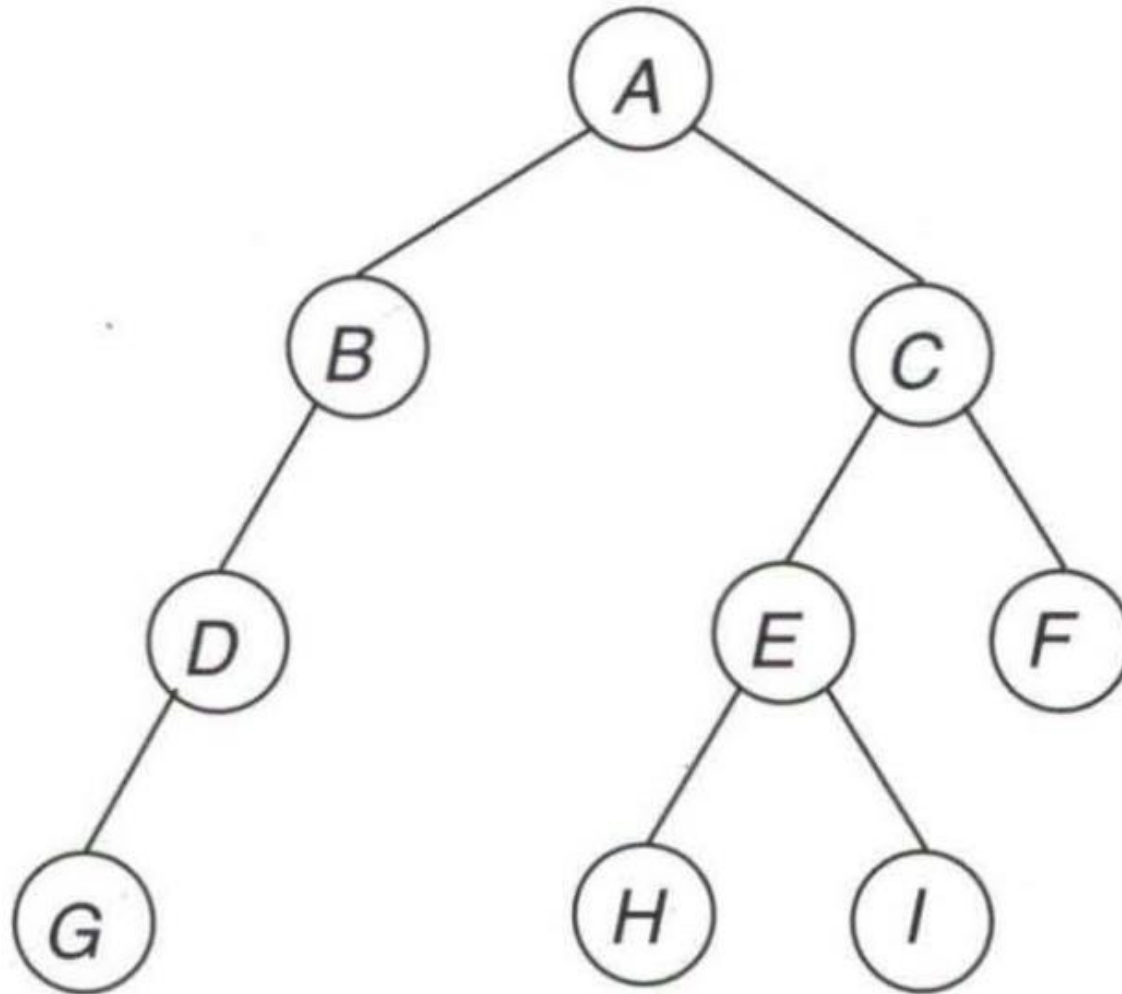
Percurso em Árvores Binárias



- **Percurso em ordem simétrica:**
- ???



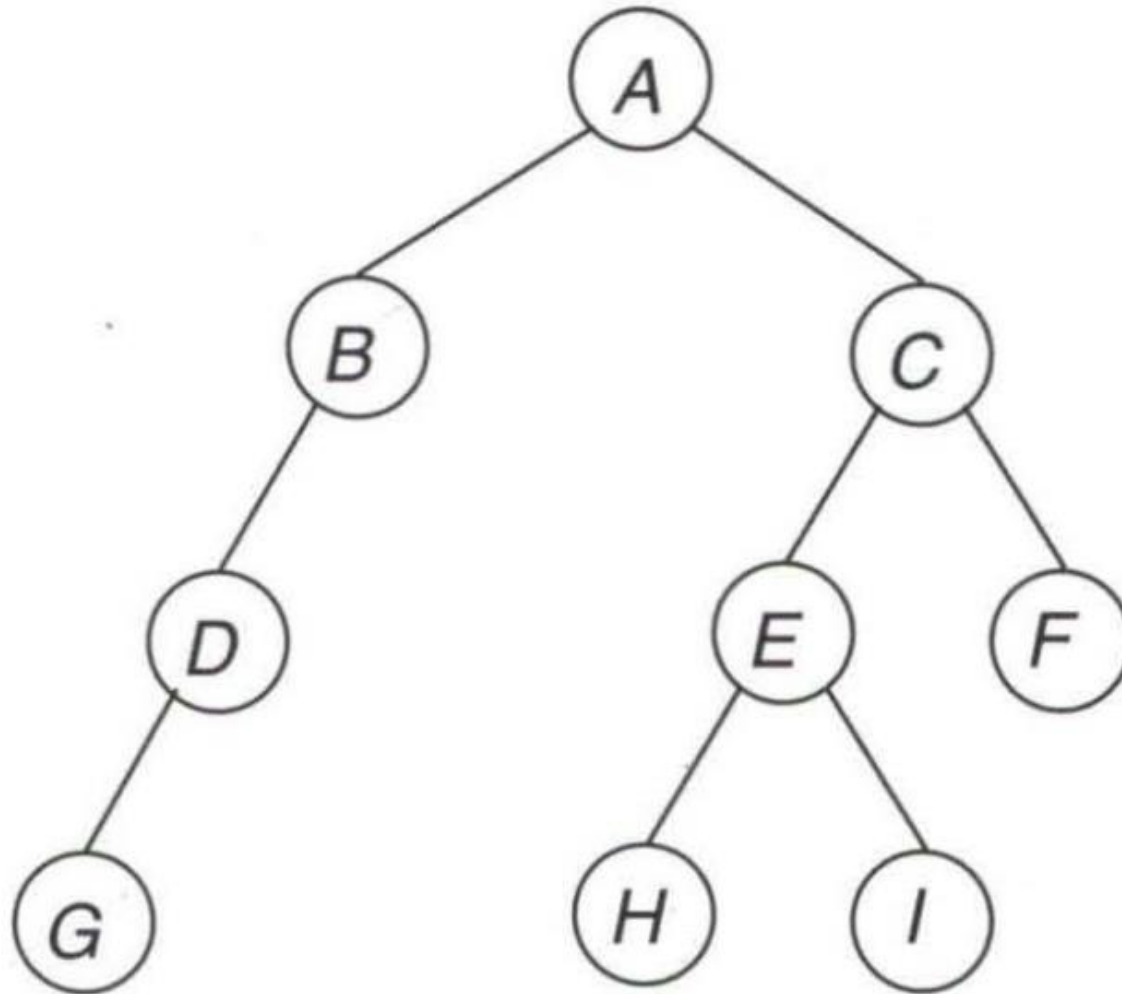
Percurso em Árvores Binárias



- **Percurso em ordem simétrica:**
- **D**



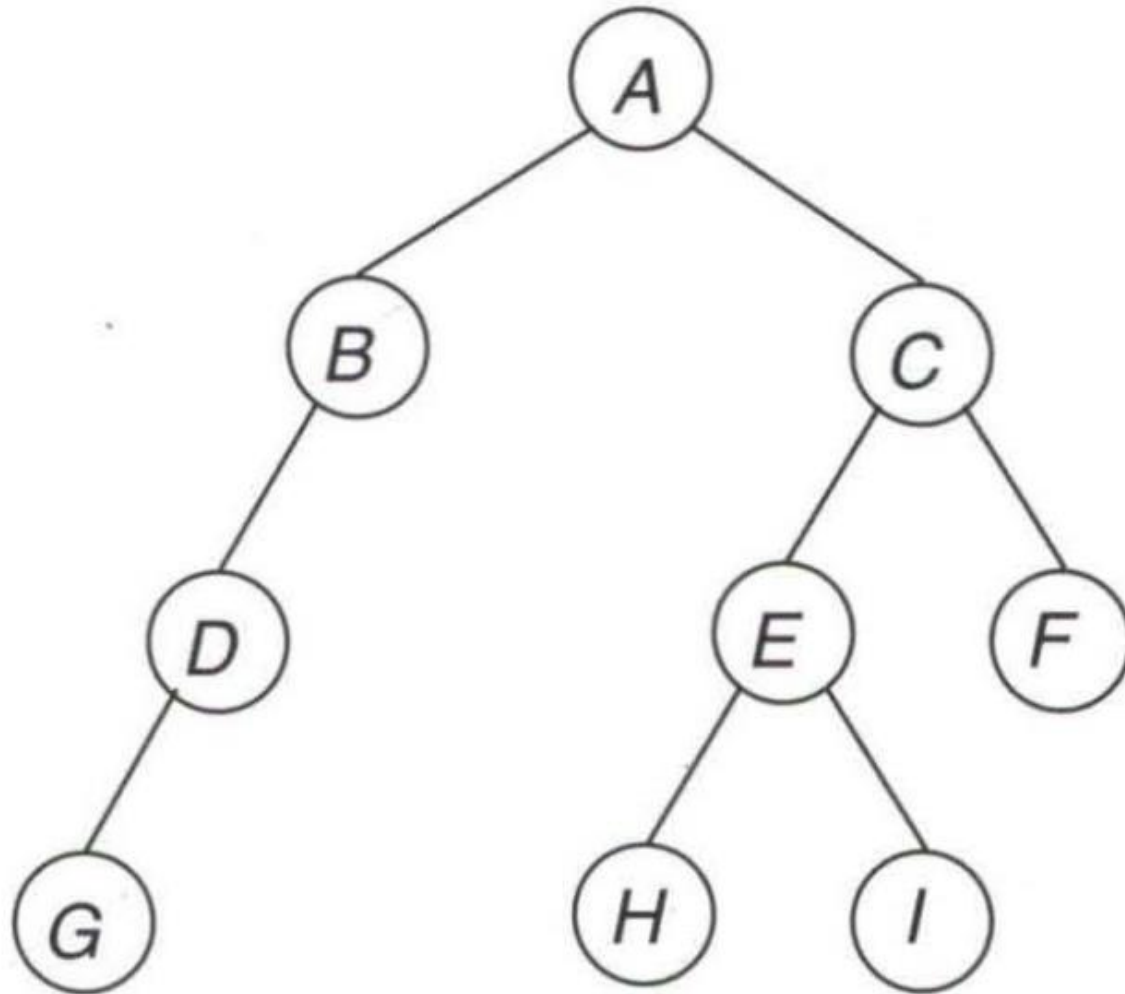
Percurso em Árvores Binárias



- **Percurso em ordem simétrica:**
- **D G**



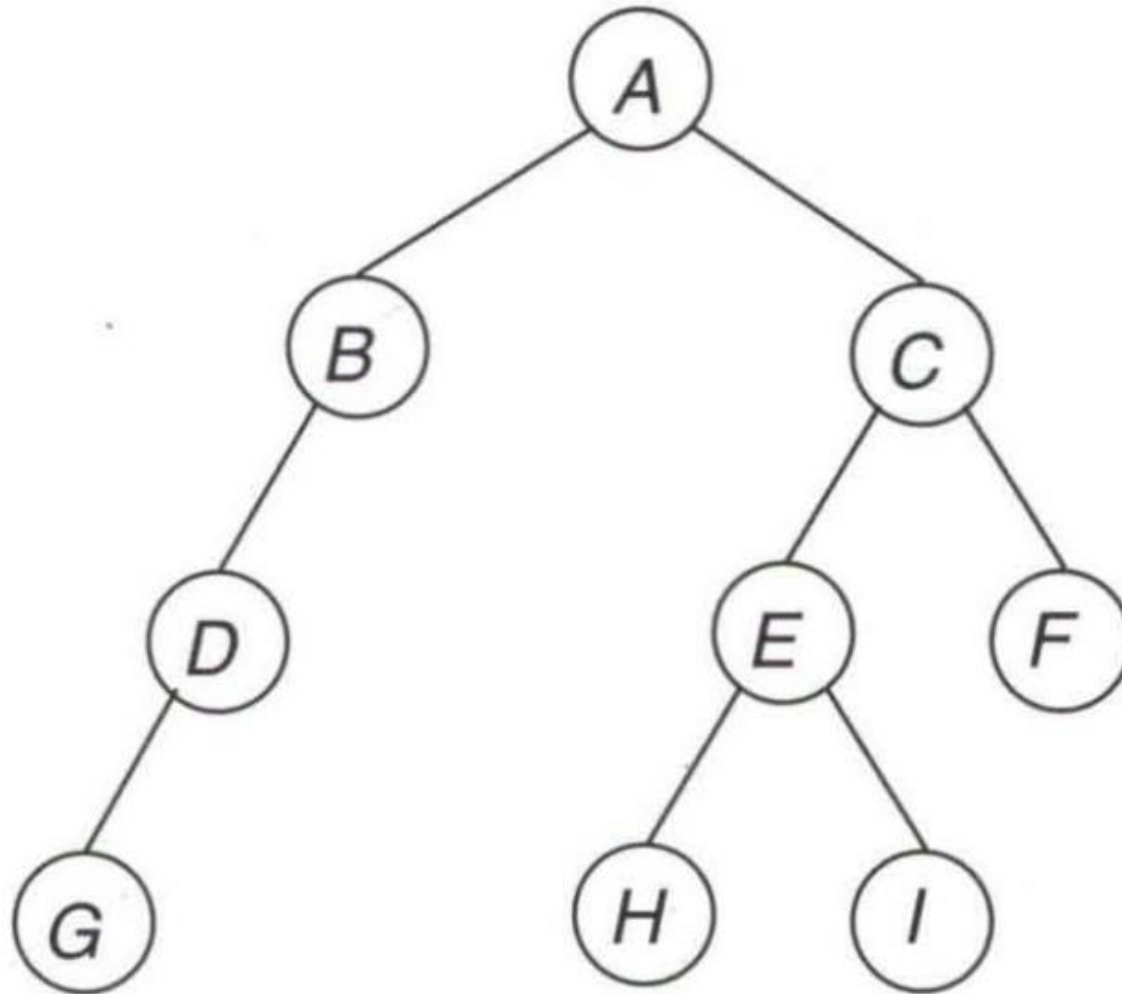
Percurso em Árvores Binárias



- Percurso em ordem simétrica:
- **D G B**



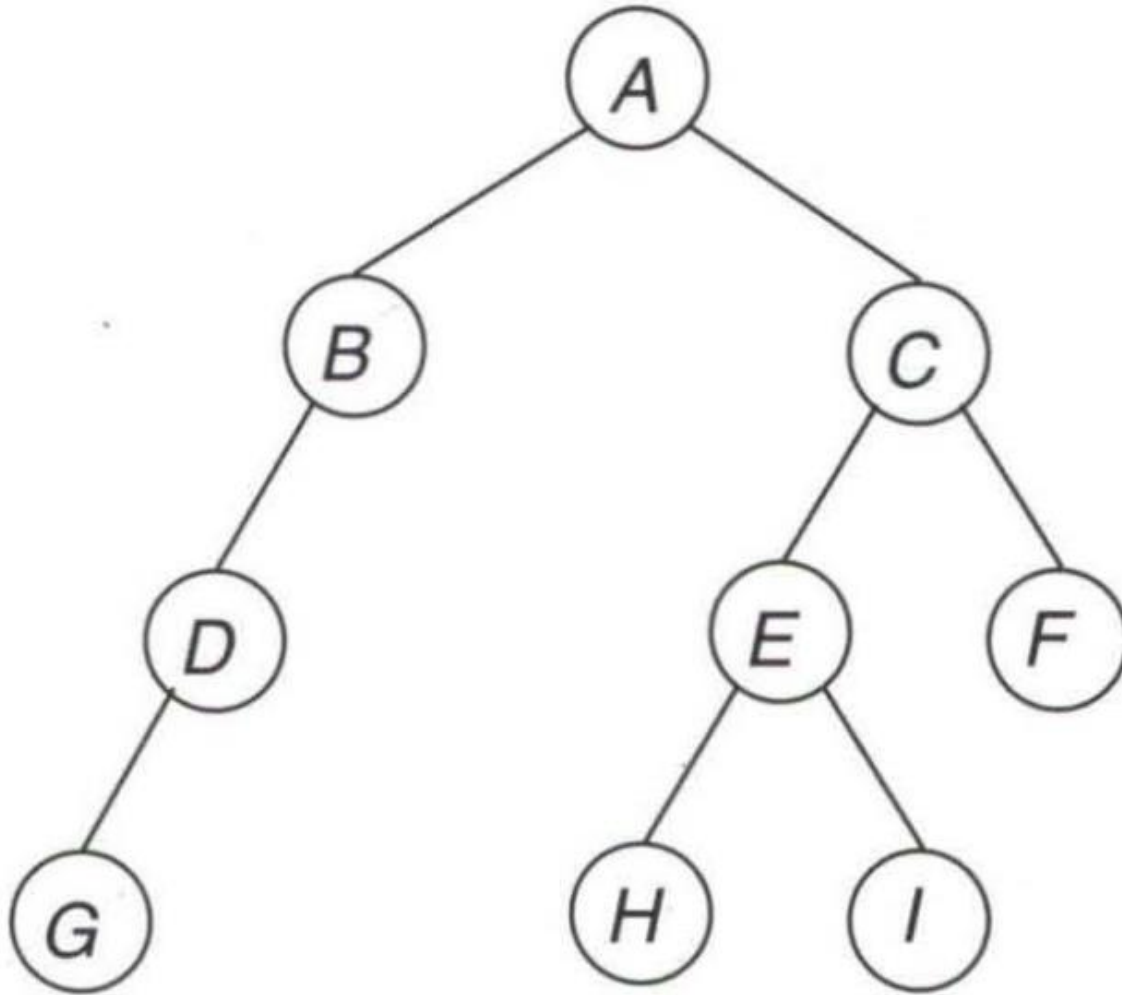
Percurso em Árvores Binárias



- Percurso em ordem simétrica:
- **D G B A**



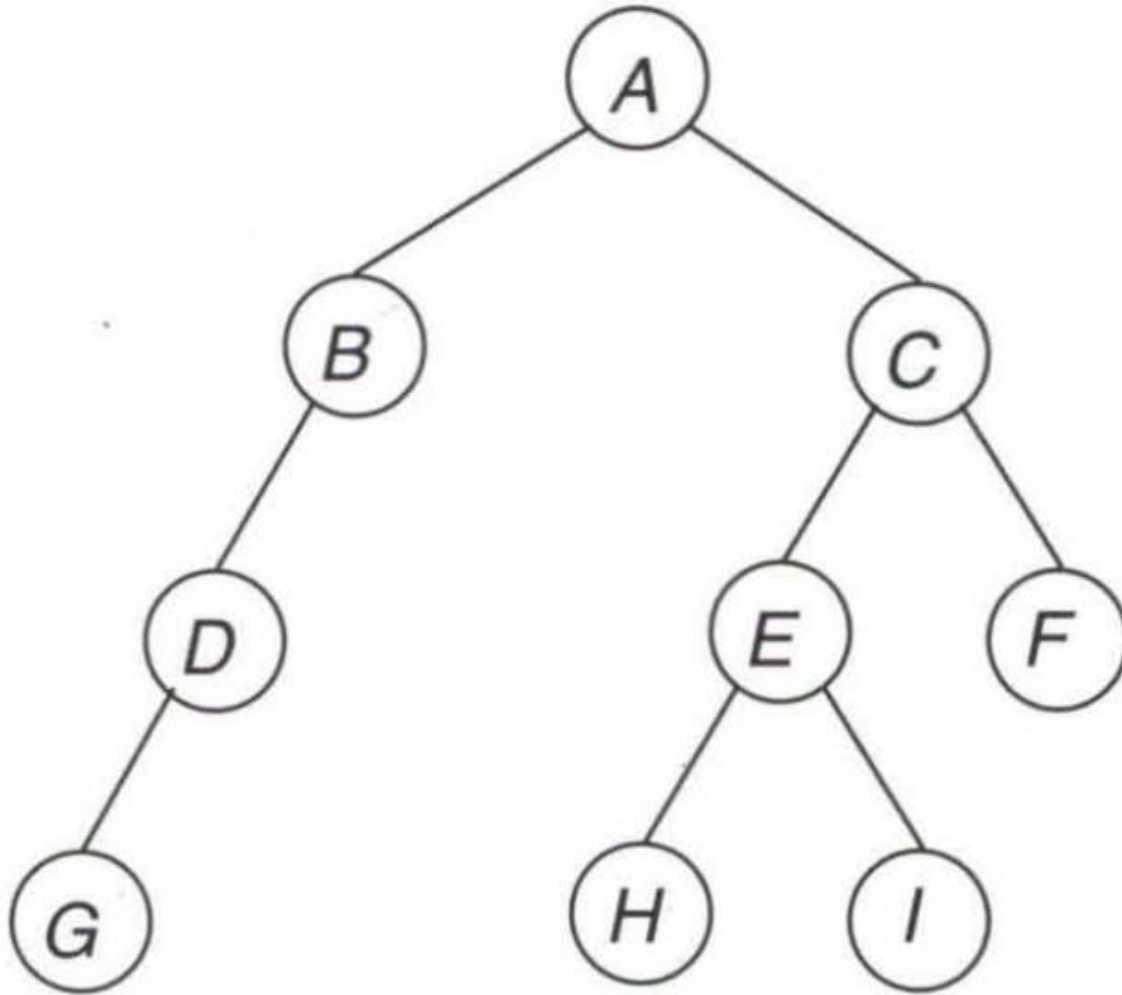
Percurso em Árvores Binárias



- Percurso em ordem simétrica:
- **D G B A H**



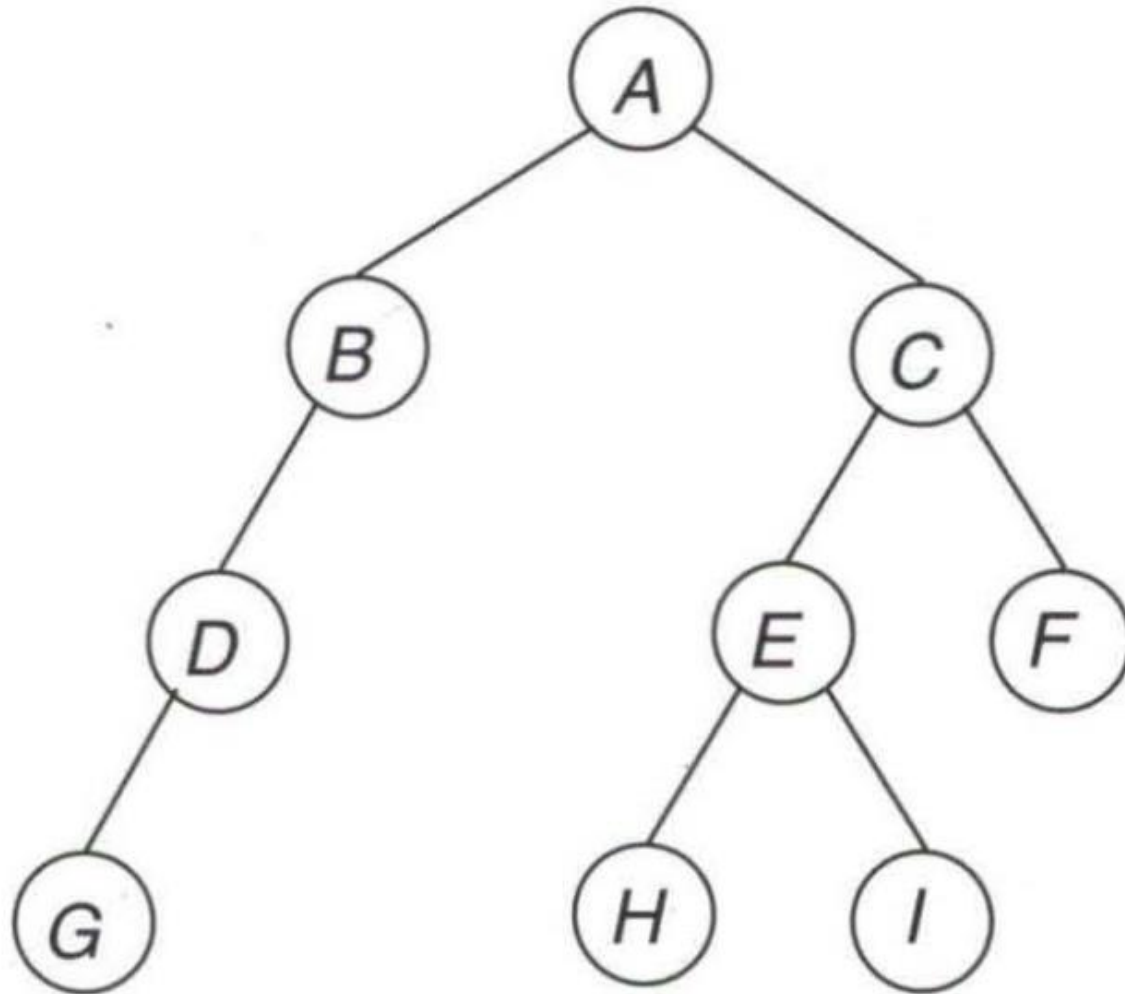
Percurso em Árvore Binária



- Percurso em ordem simétrica:
- **D G B A H E**



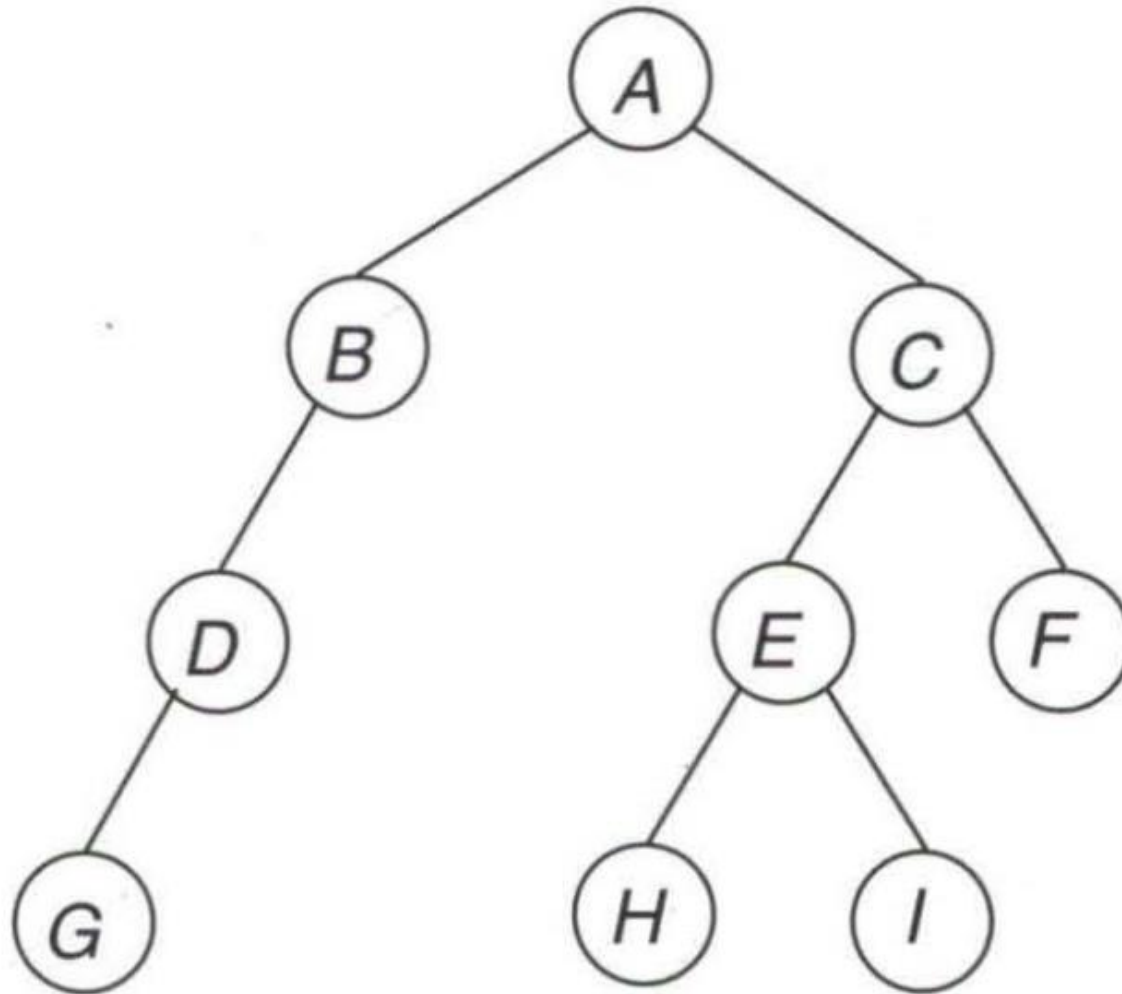
Percurso em Árvores Binárias



- Percurso em ordem simétrica:
- **D G B A H E I**



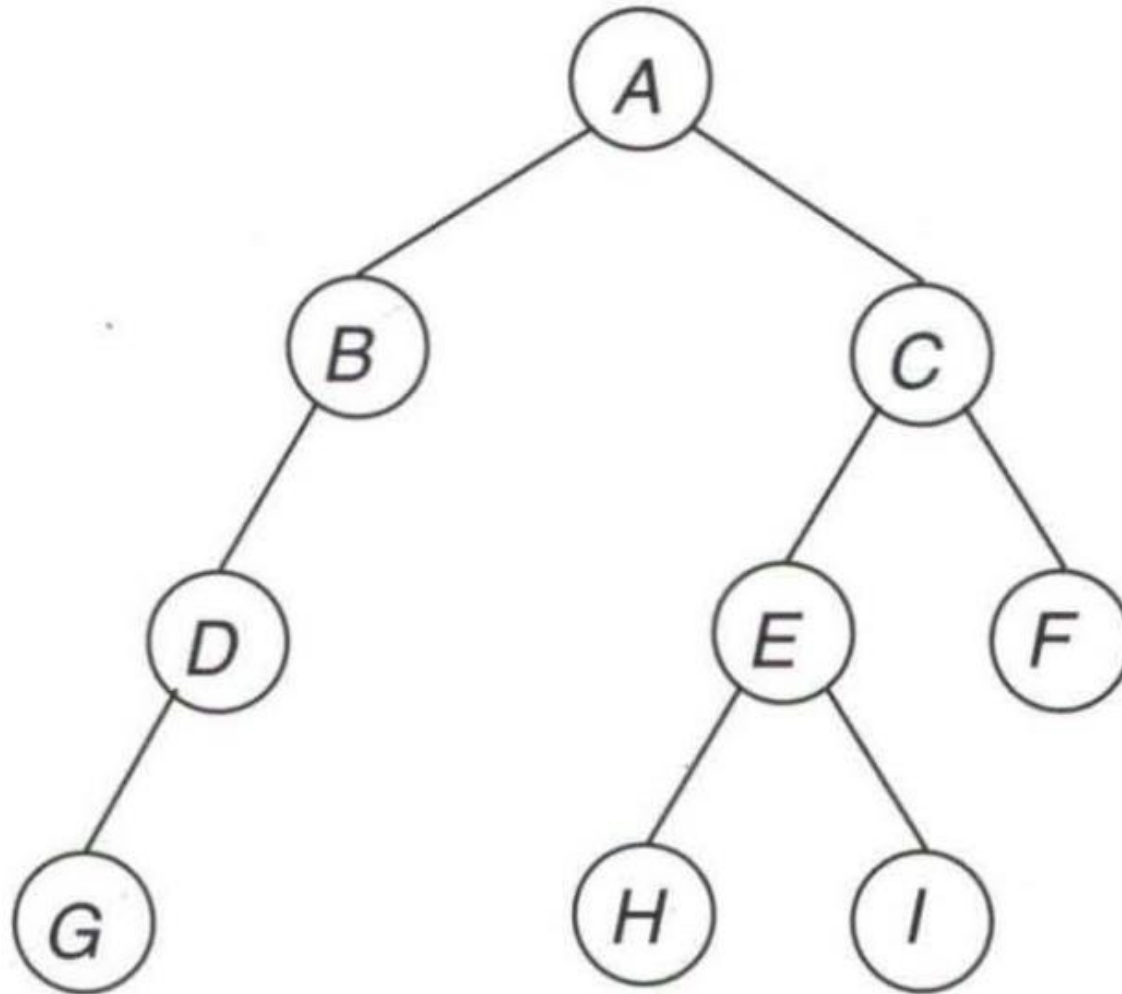
Percurso em Árvores Binárias



- Percurso em ordem simétrica:
- **D G B A H E I C**



Percurso em Árvores Binárias



- Percurso em ordem simétrica:
- **D G B A H E I C F**

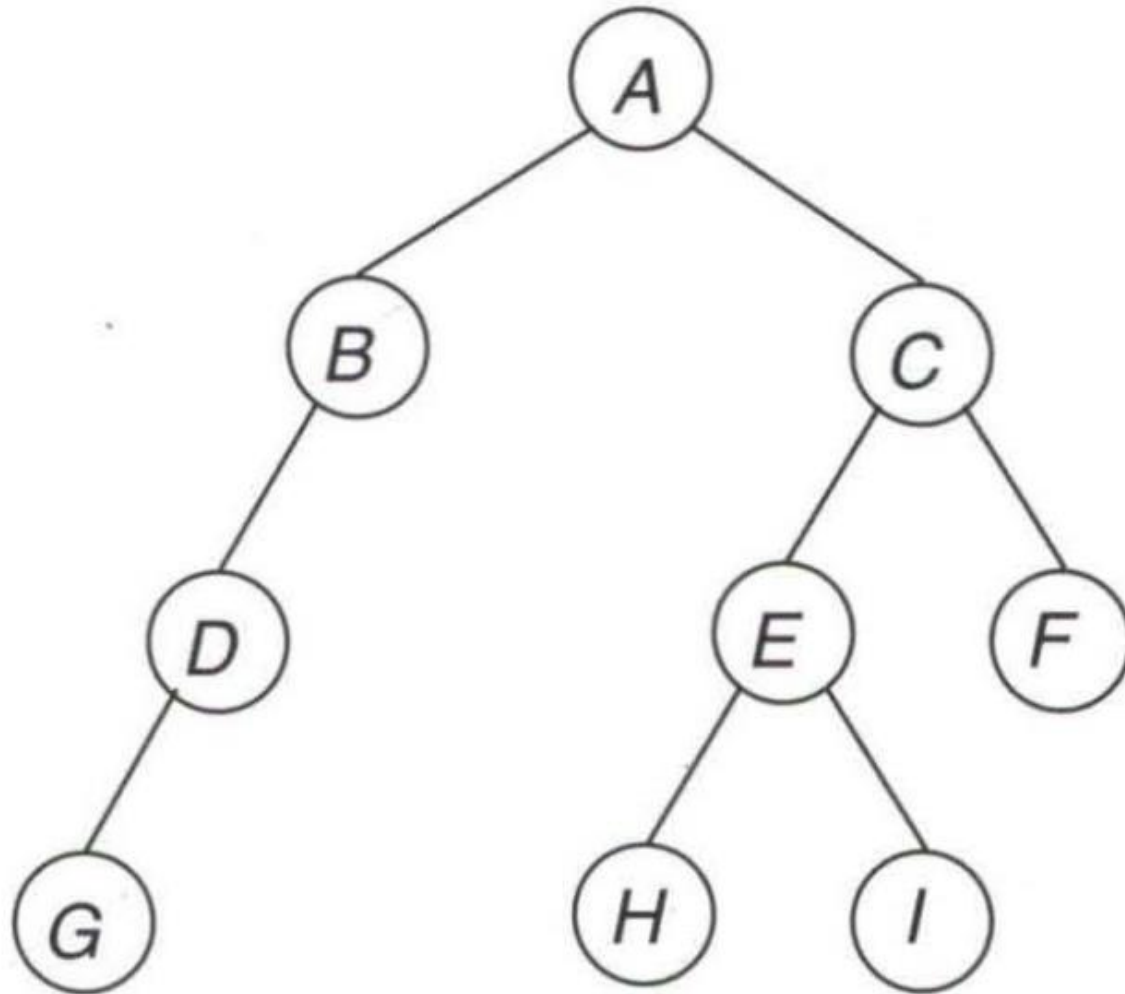


Percurso em Árvores Binárias

- Percurso em **pós-ordem**:
 - Percorrer sua sub árvore esquerda, em **pós-ordem**;
 - Percorrer sua sub árvore direita, **pós-ordem**;
 - Visitar a raiz.



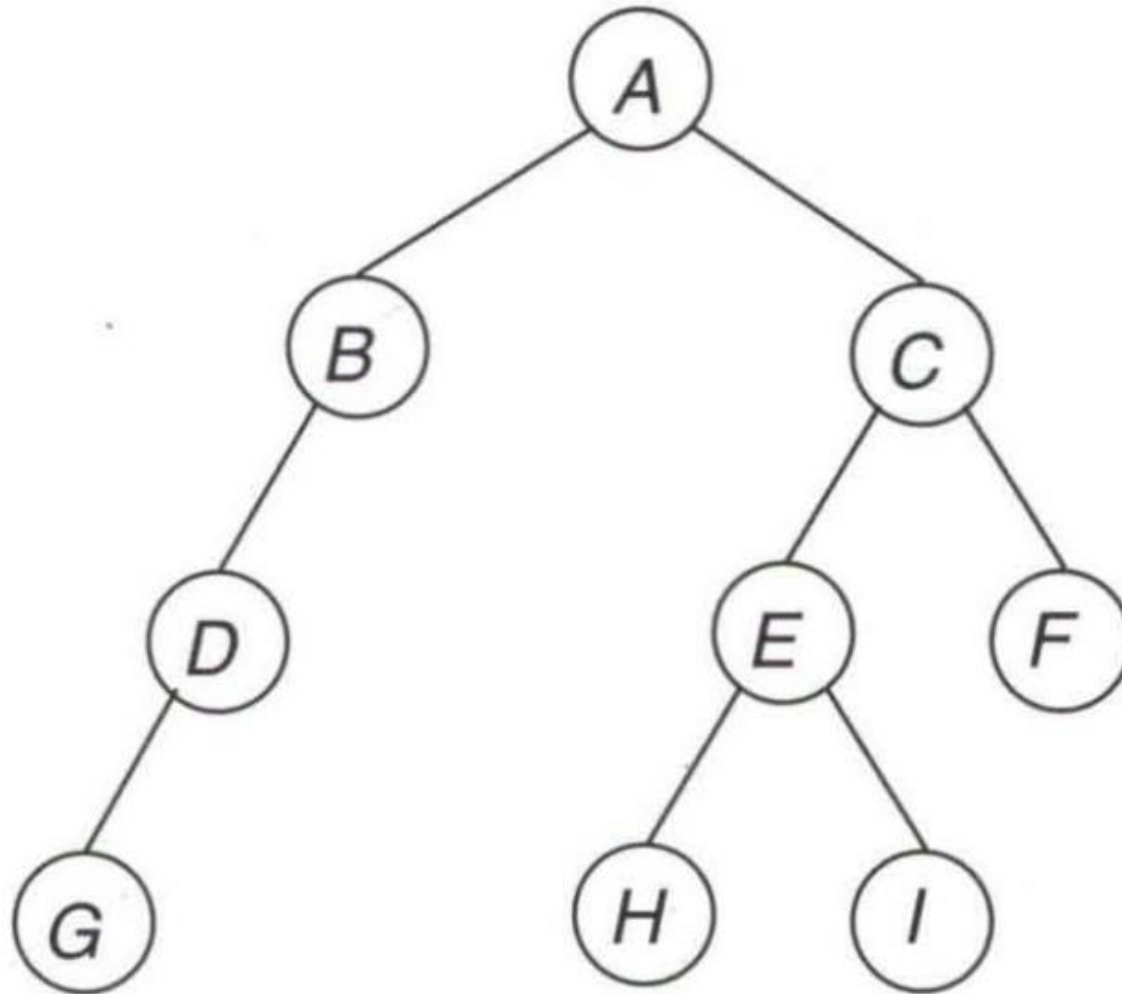
Percurso em Árvores Binárias



- **Percurso em pós-ordem :**
- ???



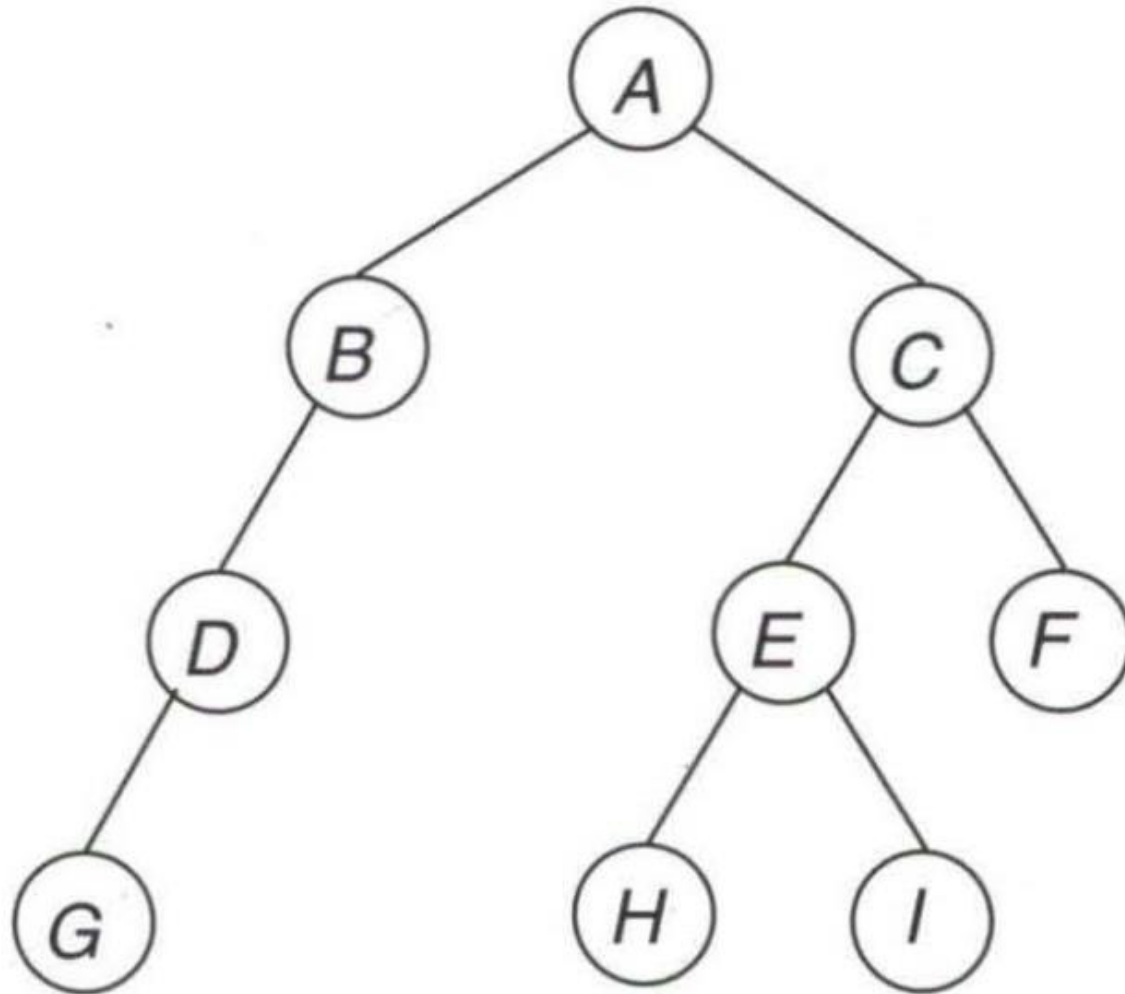
Percurso em Árvores Binárias



- **Percurso em pós-ordem :**
- **G**



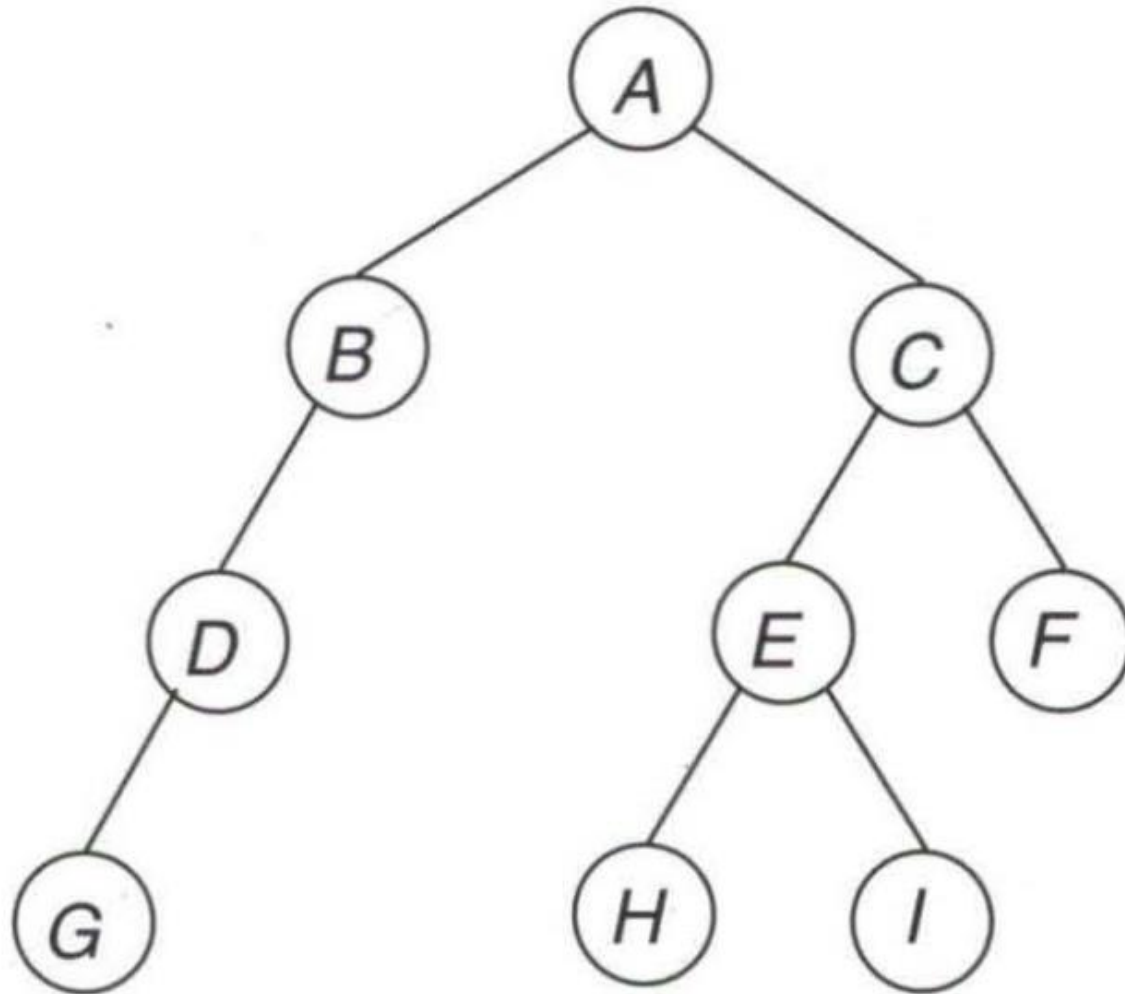
Percurso em Árvores Binárias



- **Percurso em pós-
ordem :**
- **G D**



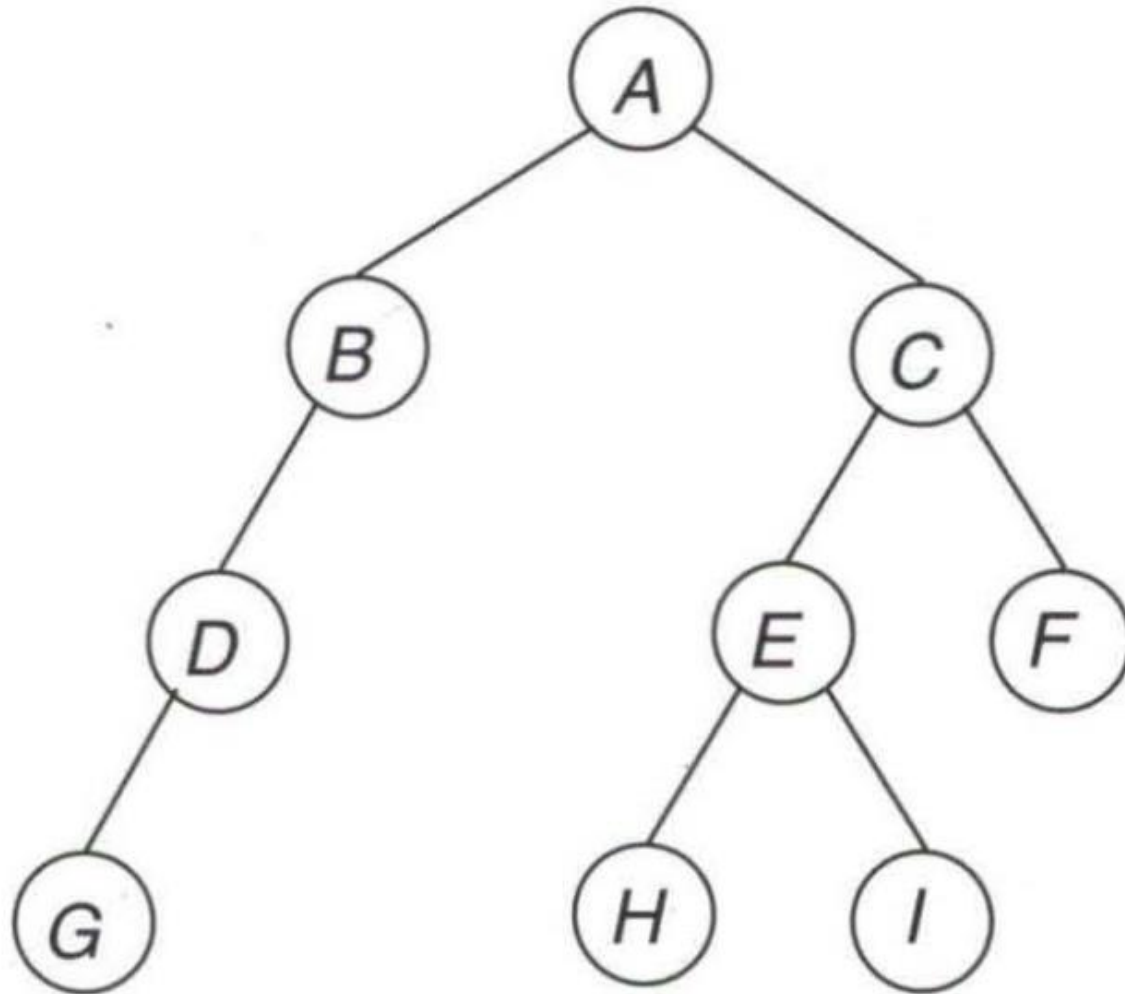
Percurso em Árvores Binárias



- **Percurso em pós-ordem :**
- **G D B**



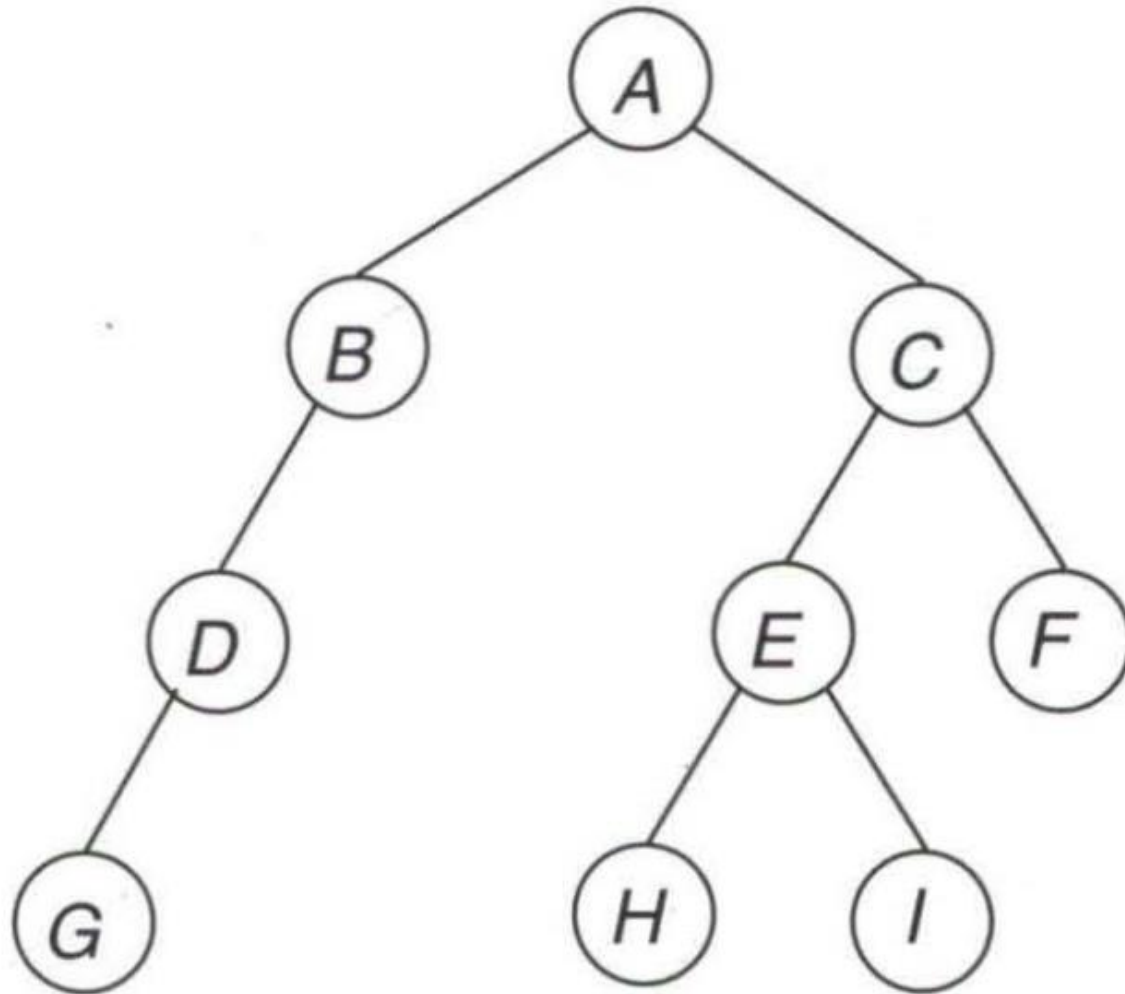
Percurso em Árvores Binárias



- Percurso em pós-
ordem :
- **G D B H**



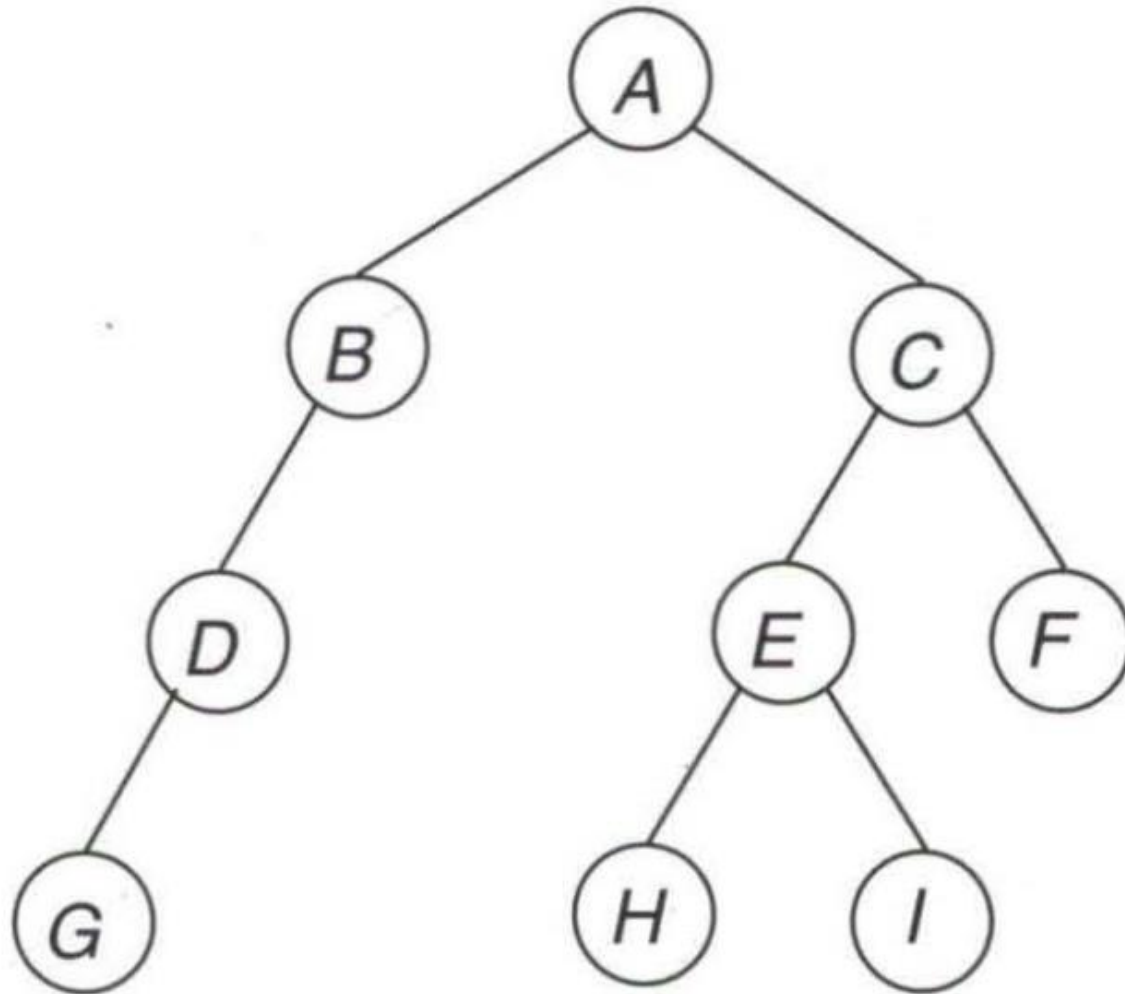
Percurso em Árvores Binárias



- Percurso em pós-
ordem :
- **G D B H I**



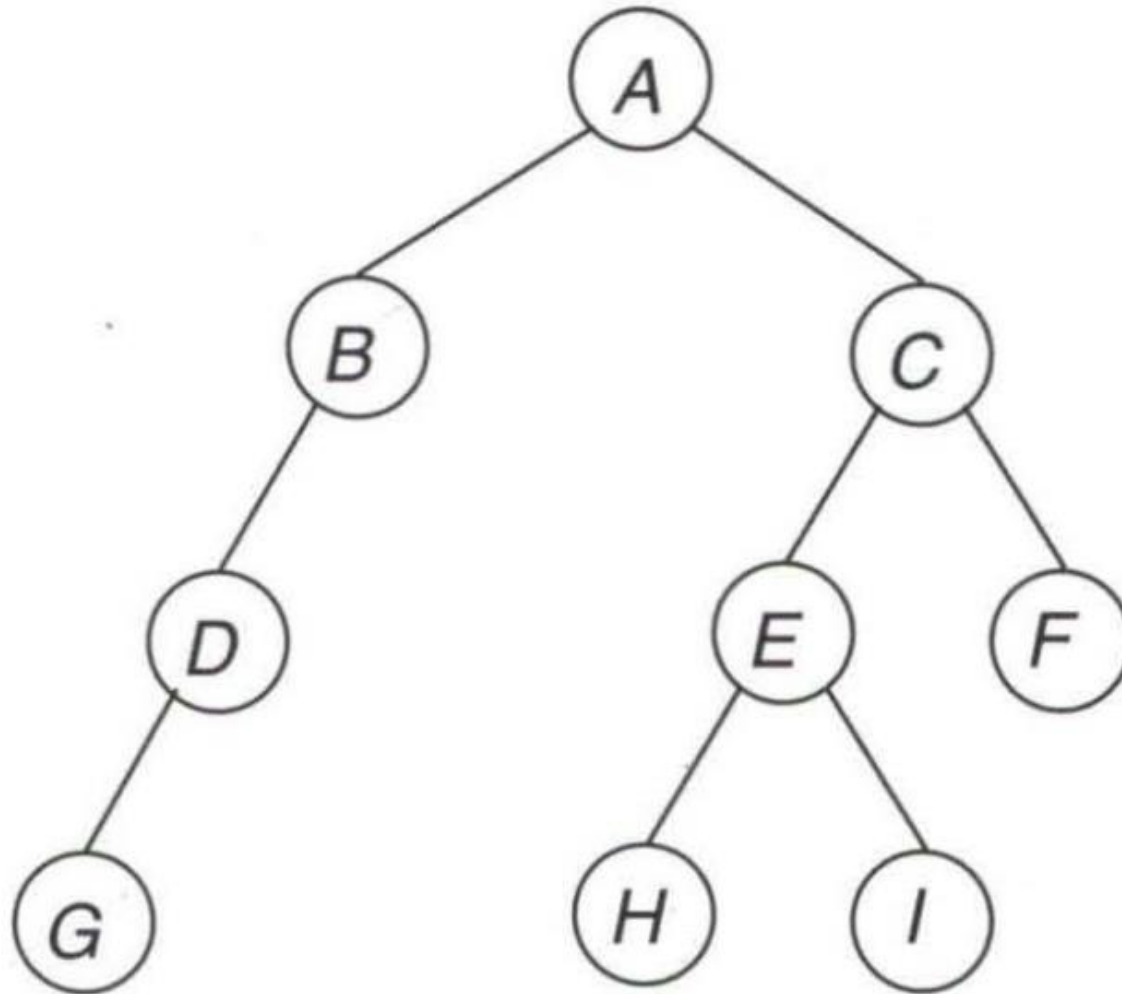
Percurso em Árvores Binárias



- Percurso em pós-ordem :
- **G D B H I E**



Percurso em Árvores Binárias

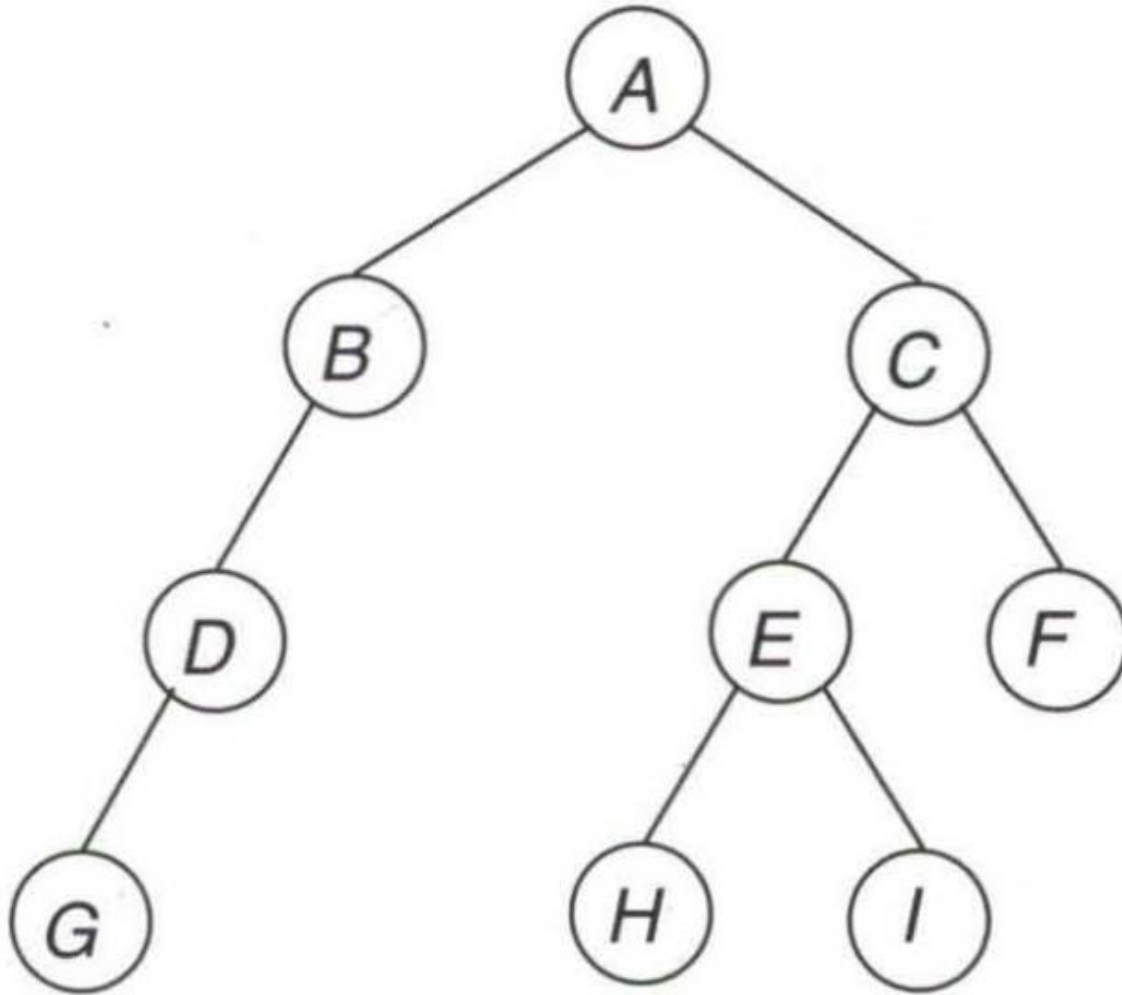


- Percurso em pós-
ordem :

- **G D B H I E F**



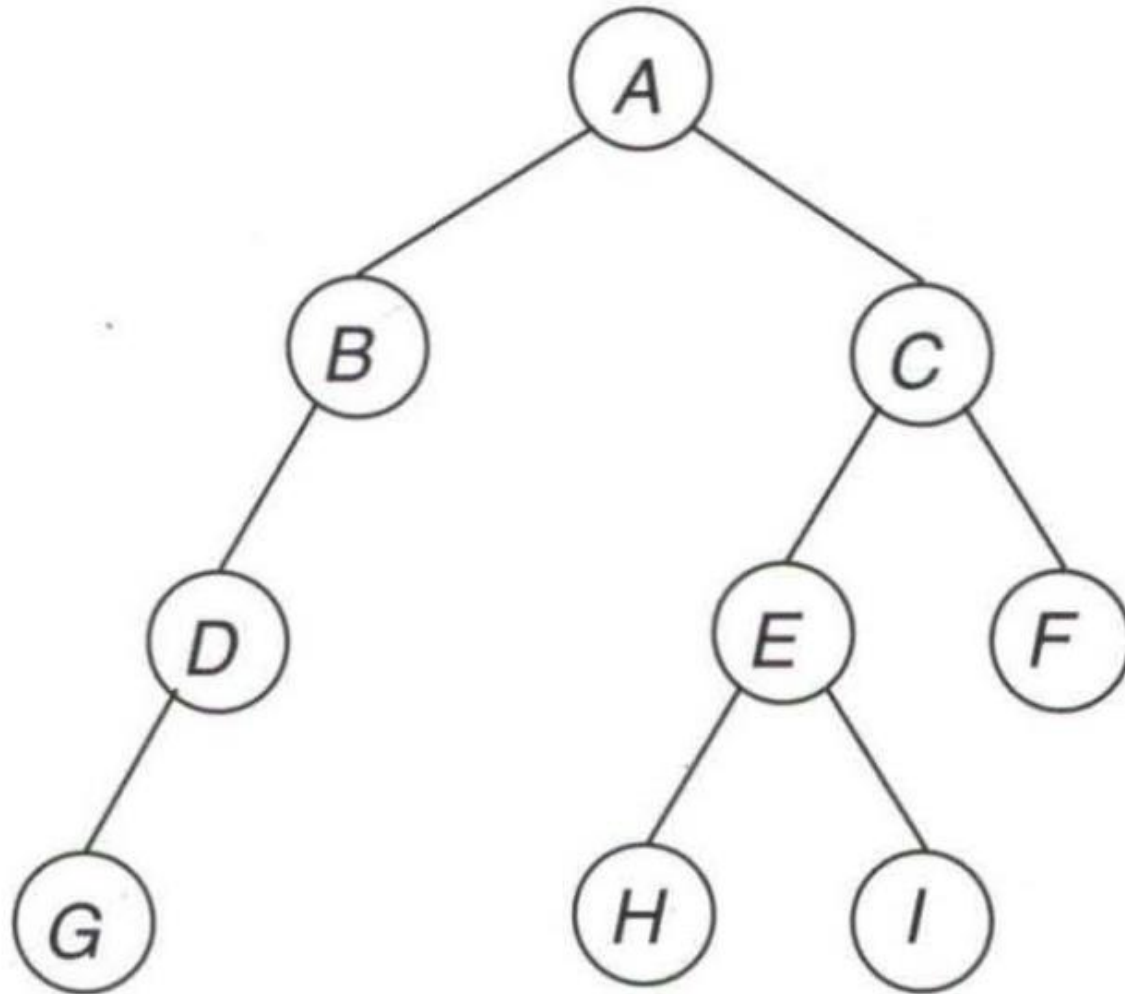
Percurso em Árvore Binária



- Percurso em pós-ordem :
- **G D B H I E F C**



Percurso em Árvores Binárias



- Percurso em pós-ordem :
- **G D B H I E F C A**



Revisão de Listas Lineares, Pilhas, Filas e Busca Binária e Introdução à Árvores

Até a próxima!