

Universidade Federal do Rio Grande do Norte
Instituto Metr pole Digital

AULA 07

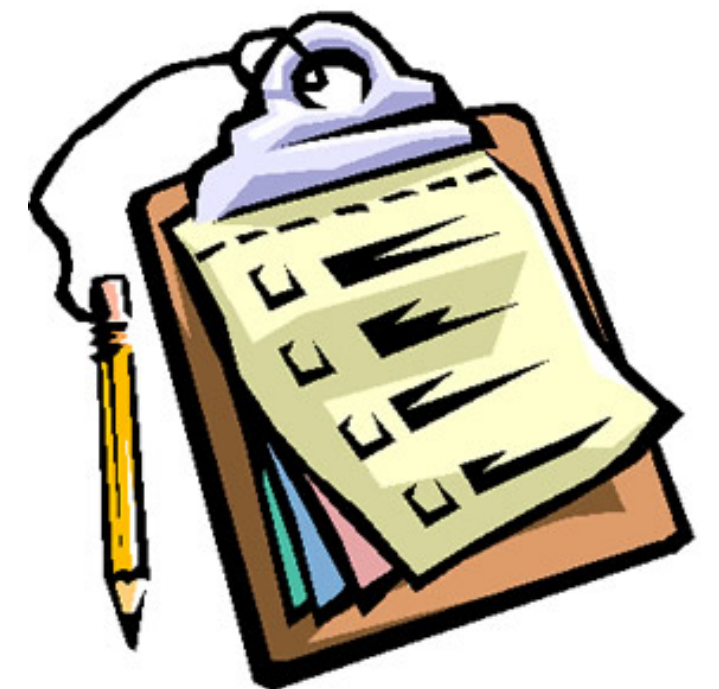
Estrutura de dados b sico I (EDB1)

Prof. Msc. Janiheryson Felipe (Felipe)

Natal, RN
2023

OBJETIVO DA AULA

- Apresentar algoritmos de ordenação;
 - Conhecer o conceito de arvores
 - Conhecer a árvore de busca binária:

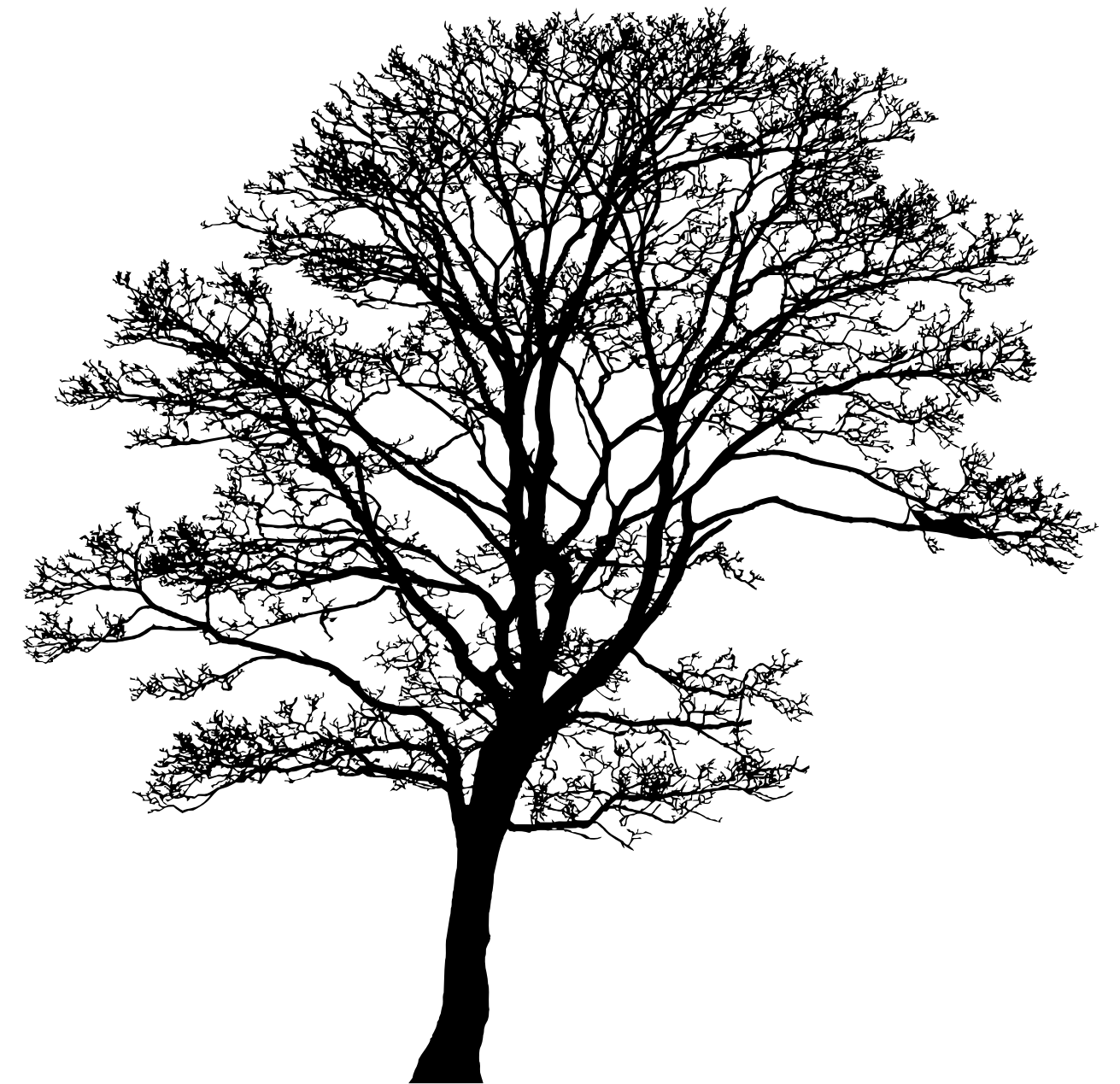




ALGORITMOS DE BUSCA

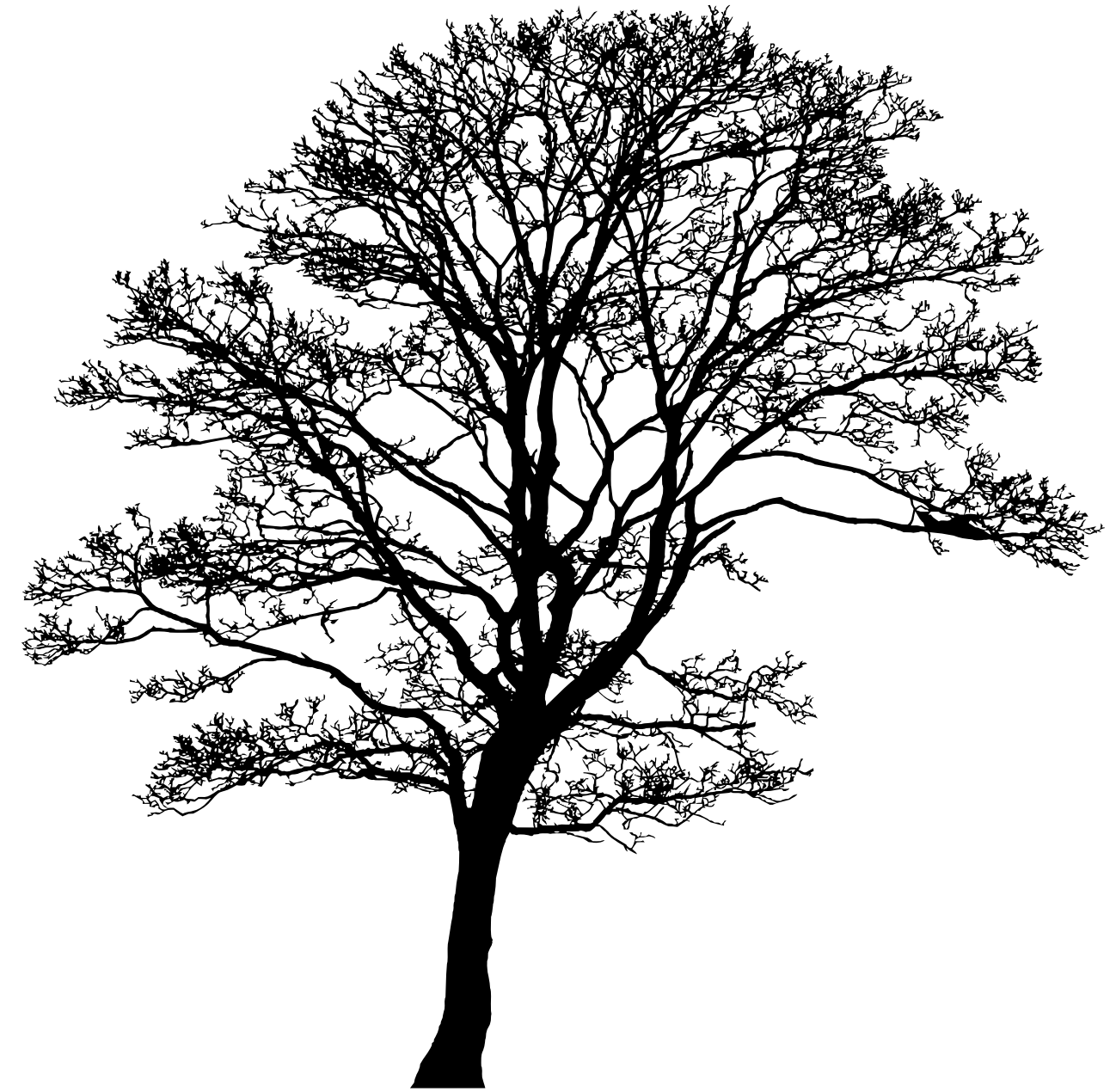
O QUE SÃO ARVORES?

Uma árvore é uma estrutura de dados fundamental em programação que consiste em um conjunto de elementos chamados de nós, conectados entre si por meio de arestas ou linhas.

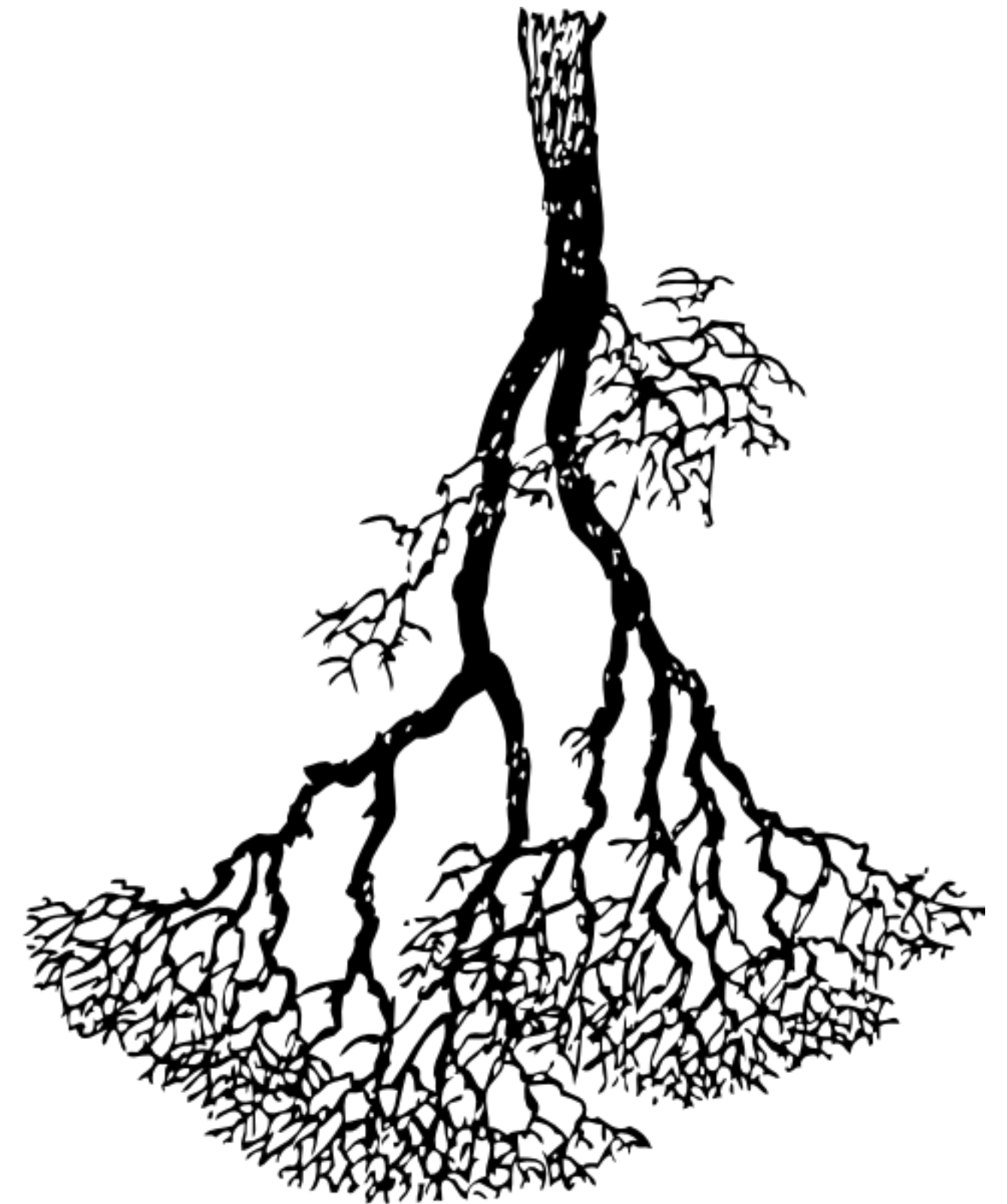
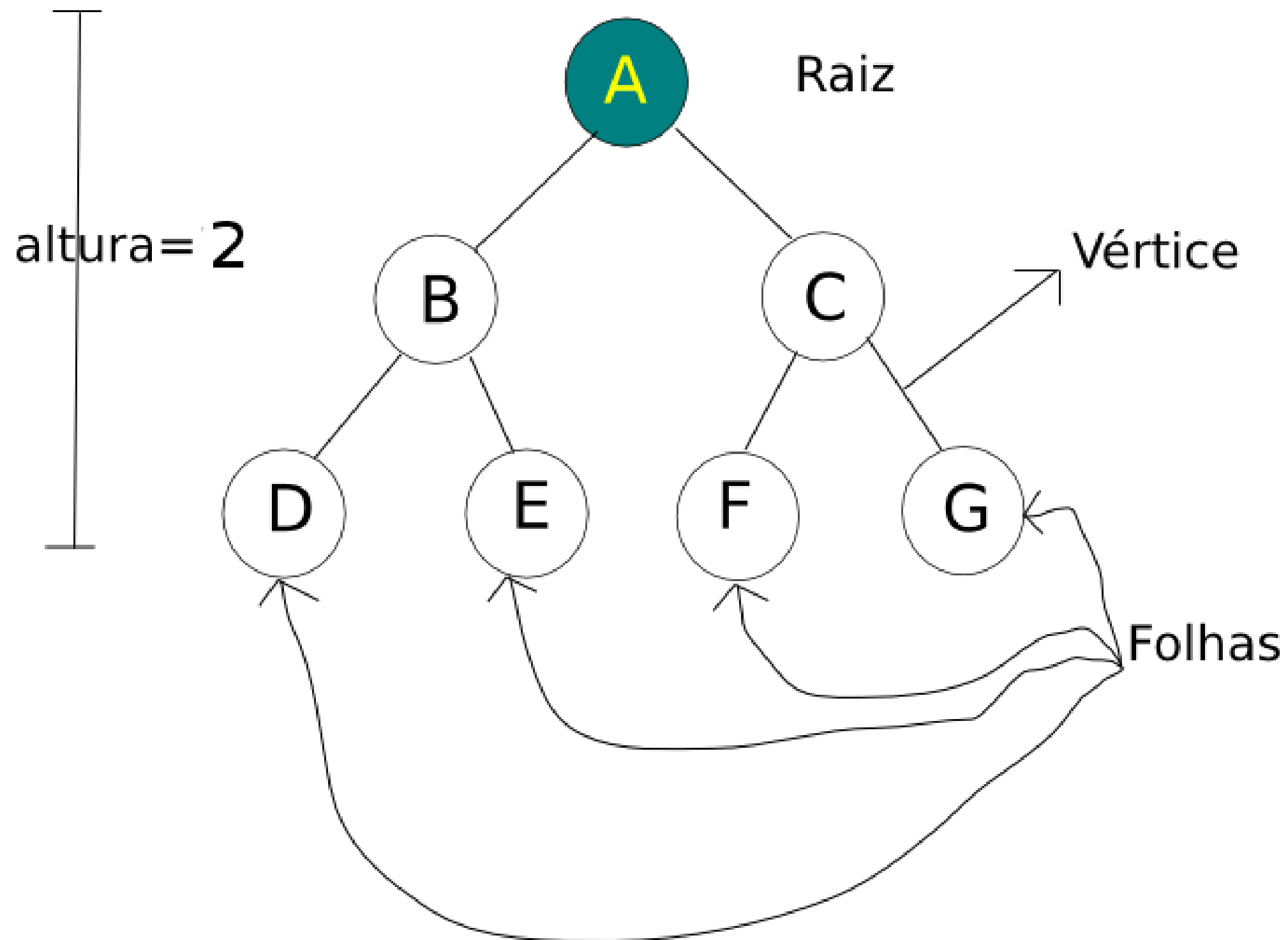


O QUE SÃO ÁRVORES?

Os nós em uma árvore têm uma hierarquia bem definida, onde cada nó pode ter um ou mais nós filhos, exceto o nó raiz, que é o nó de topo e não tem pai. Cada nó em uma árvore pode ter um número limitado de filhos, que depende do tipo de árvore



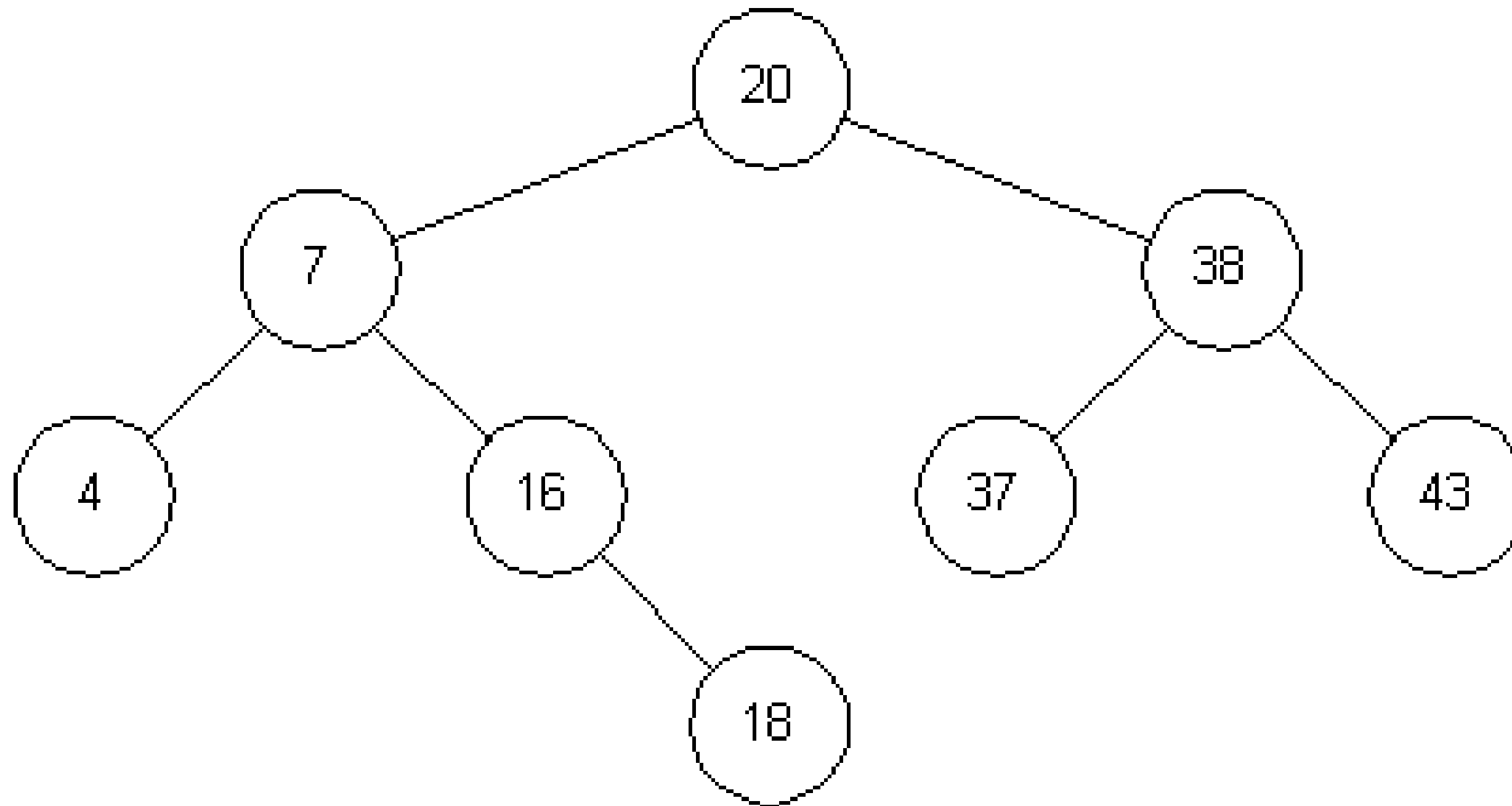
O QUE SÃO ARVORES?



ELEMENTOS DE UMA ARVORE

- **Nó:** um elemento da árvore que contém um valor de dados e pode ter zero, um ou vários nós filhos.
- **Raiz:** o nó superior da árvore que não tem um nó pai.
- **Folha:** um nó sem filhos na árvore.
- **Filho:** um nó que é diretamente conectado a outro nó abaixo dele na hierarquia.
- **Pai:** o nó que está diretamente acima de um nó na hierarquia da árvore.
- **Caminho:** uma sequência de nós conectados por arestas que conectam um nó a outro nó na árvore.
- **Altura:** a distância entre a raiz e a folha mais distante na árvore.

ELEMENTOS DE UMA ARVORE

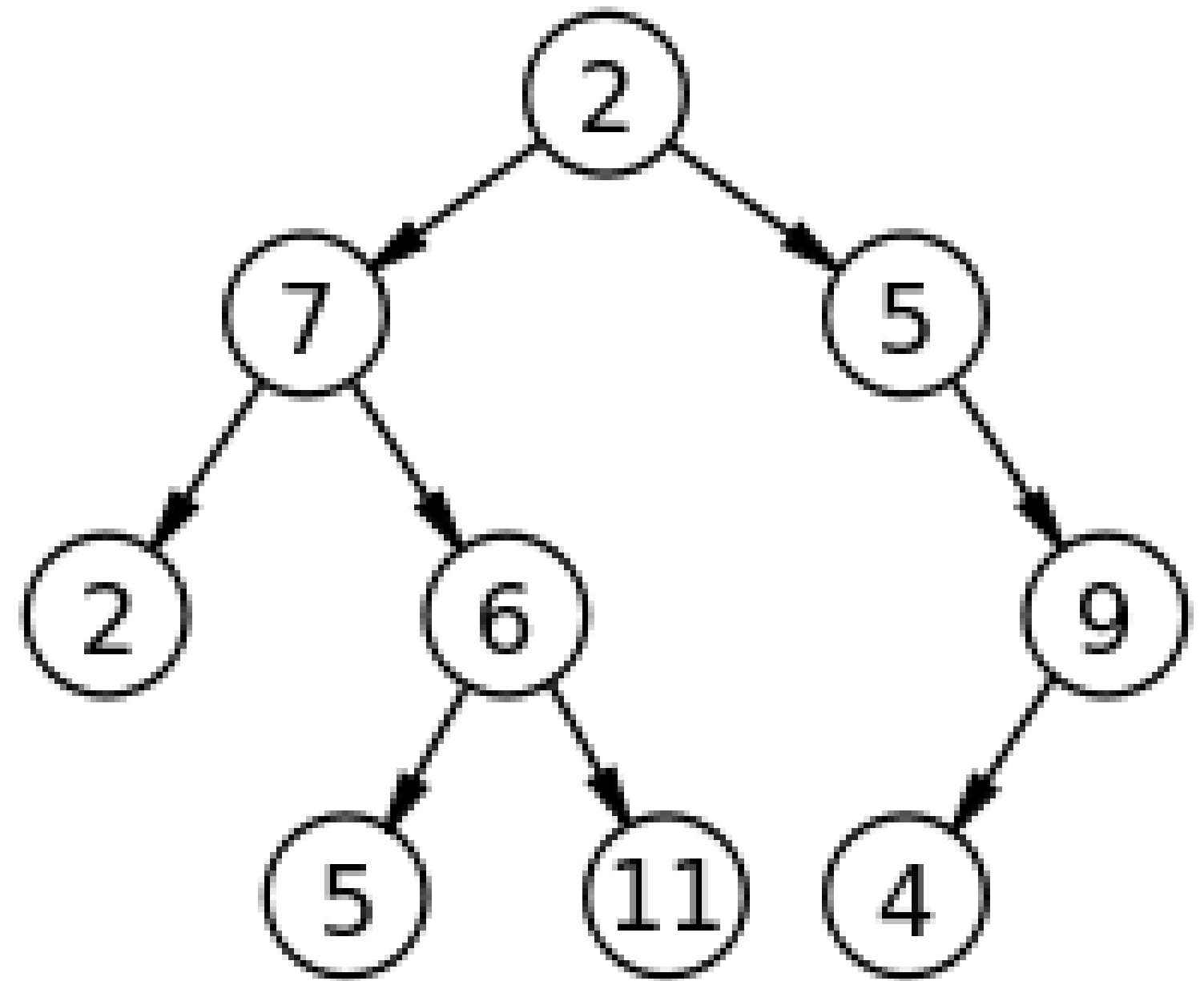


ÁRVORE BINÁRIA DE BUSCA

Árvores binárias são um tipo especial de árvore em que cada nó pode ter no máximo dois filhos, que são chamados de filho esquerdo e filho direito. Cada filho esquerdo é sempre menor ou igual ao pai, e cada filho direito é sempre maior que o pai, em termos de uma ordem específica definida para a árvore. Isso torna as árvores binárias úteis para implementar algoritmos de busca e ordenação eficientes.

ARVORE BINARIA DE BUSCA

As árvores binárias são amplamente utilizadas em algoritmos de busca, como a busca binária, que é uma técnica eficiente para encontrar um elemento em um conjunto ordenado



PERCURSO EM UMA ARVORE BINARIA

Existem três formas comuns de percurso (ou travessia) em uma árvore binária:

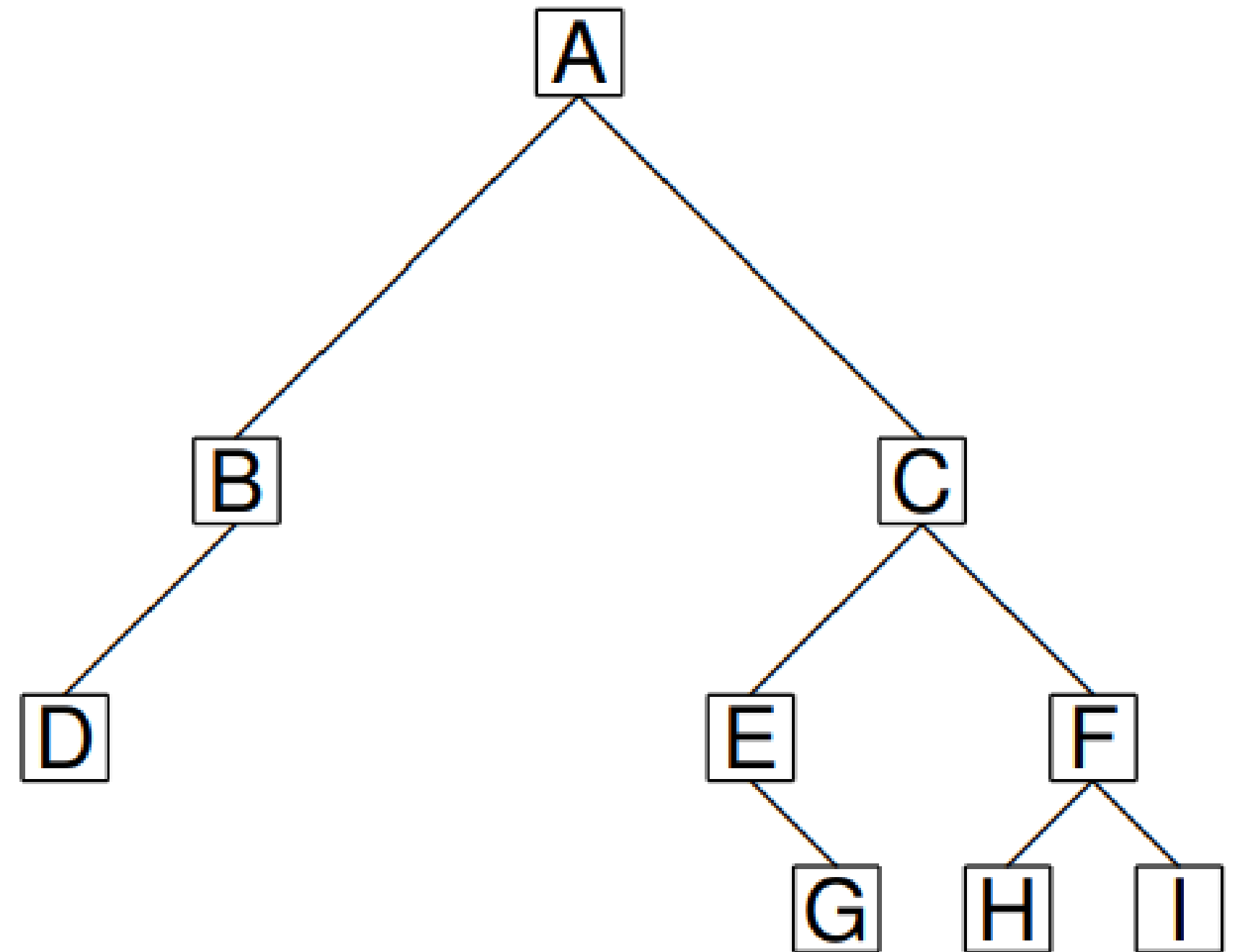
- **Pré-ordem (preorder):** nesse percurso, o nó raiz é visitado primeiro, seguido pela subárvore esquerda e, em seguida, pela subárvore direita. Em outras palavras, a ordem de percurso é: raiz, esquerda, direita.

PERCURSO EM UMA ARVORE BINARIA

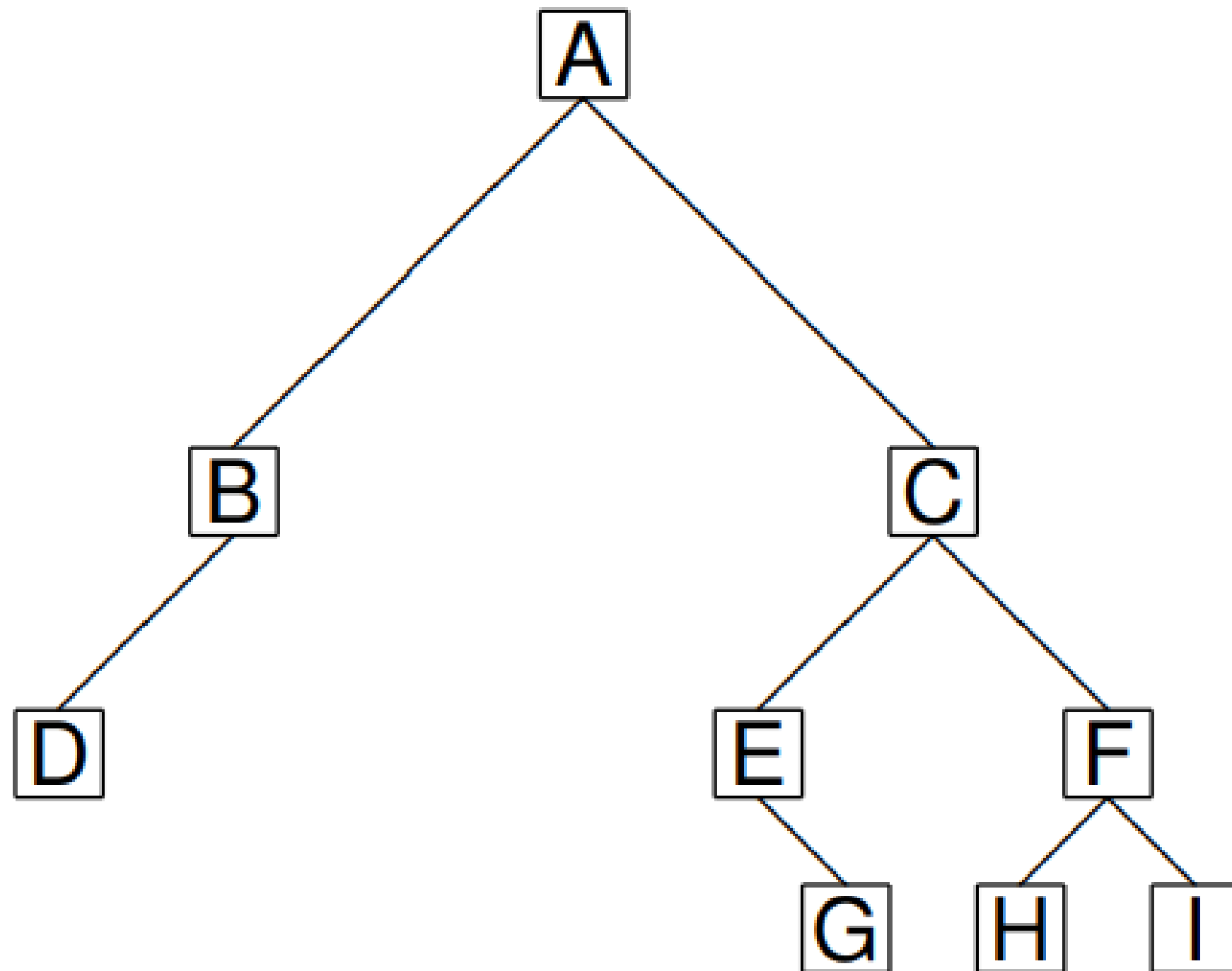
Percorrer uma árvore binária em pré-ordem:

- 1 Visitar a raiz.
- 2 Visita a esquerda.
- 3 Visita a direita.

Visitar um nó significa executar uma certa ação no nó.



PERCURSO EM UMA ARVORE BINARIA



PERCURSO EM PRÉ-ORDEM:

A B D C E G F H I

PERCURSO EM UMA ARVORE BINARIA

Existem três formas comuns de percurso (ou travessia) em uma árvore binária:

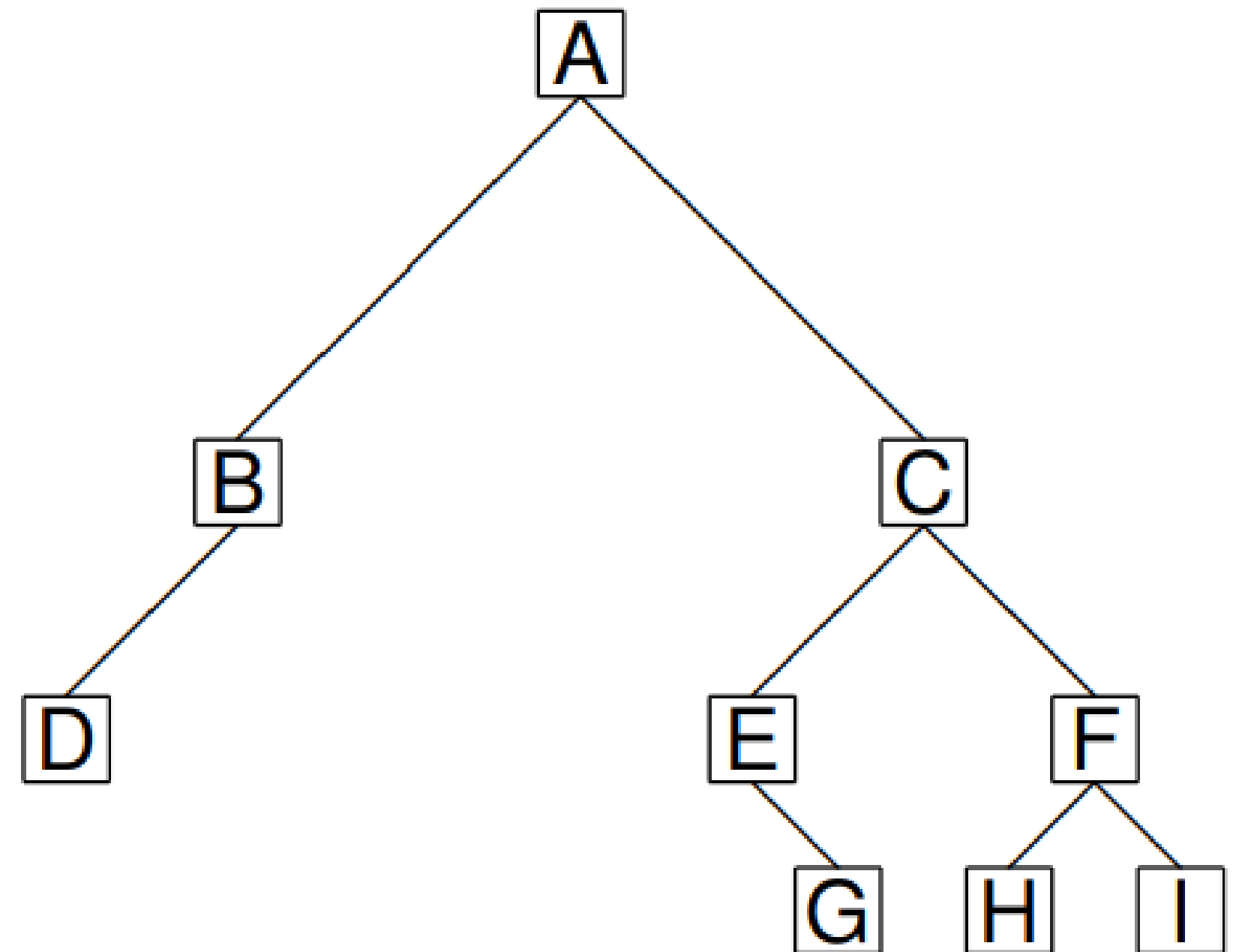
- Em ordem (inorder): nesse percurso, a subárvore esquerda é visitada primeiro, seguida pelo nó raiz e, em seguida, pela subárvore direita. Em outras palavras, a ordem de percurso é: esquerda, raiz, direita.

PERCURSO EM UMA ARVORE BINARIA

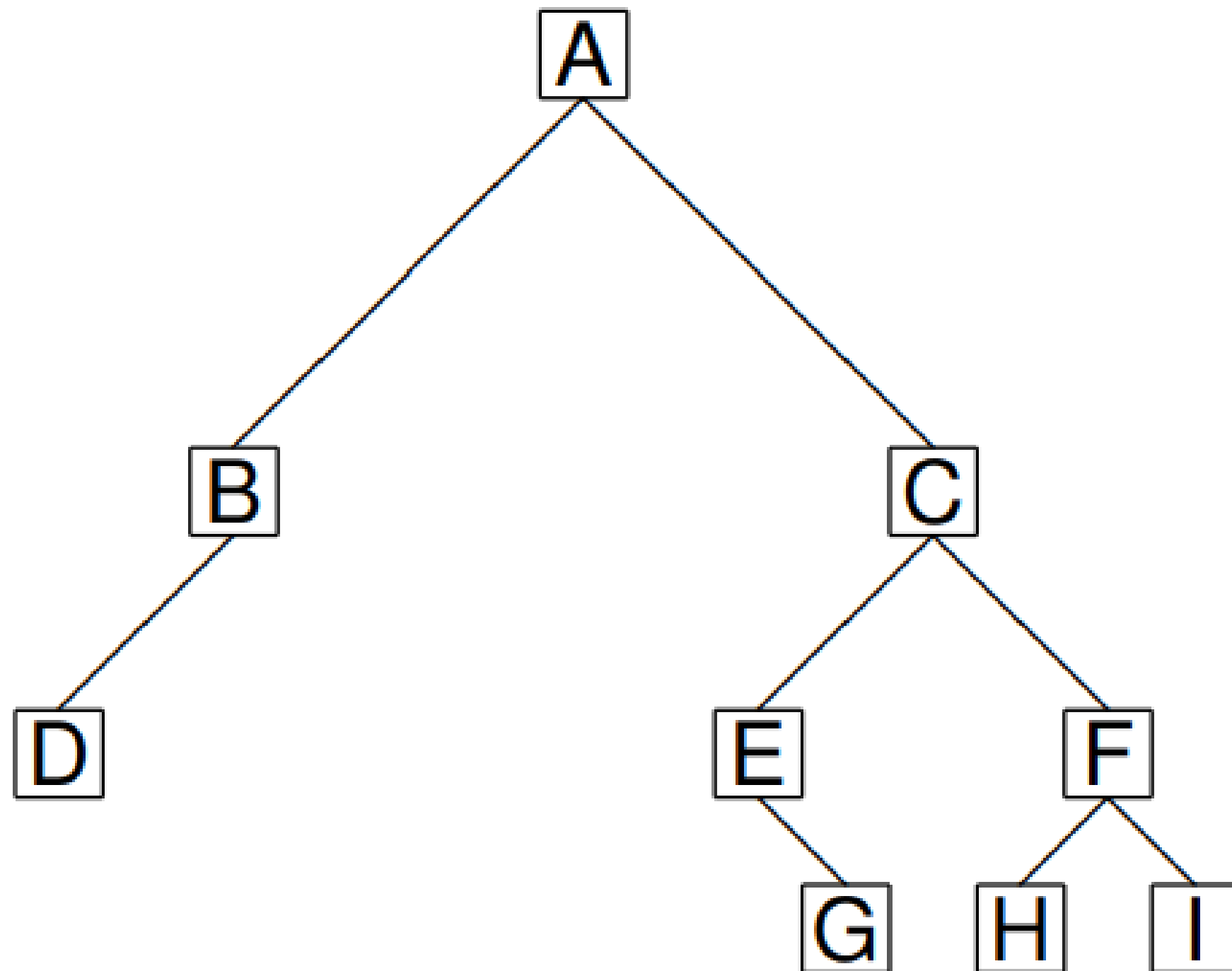
Percorrer uma árvore binária em in-ordem:

- 1 Visita a esquerda.
- 2 Vistar a raiz.
- 3 Visita a direita

É conhecida também pelo nome de ordem simétrica.



PERCURSO EM UMA ARVORE BINARIA



PERCURSO EM IN-ORDEM:

D B A E G C H F I

PERCURSO EM UMA ARVORE BINARIA

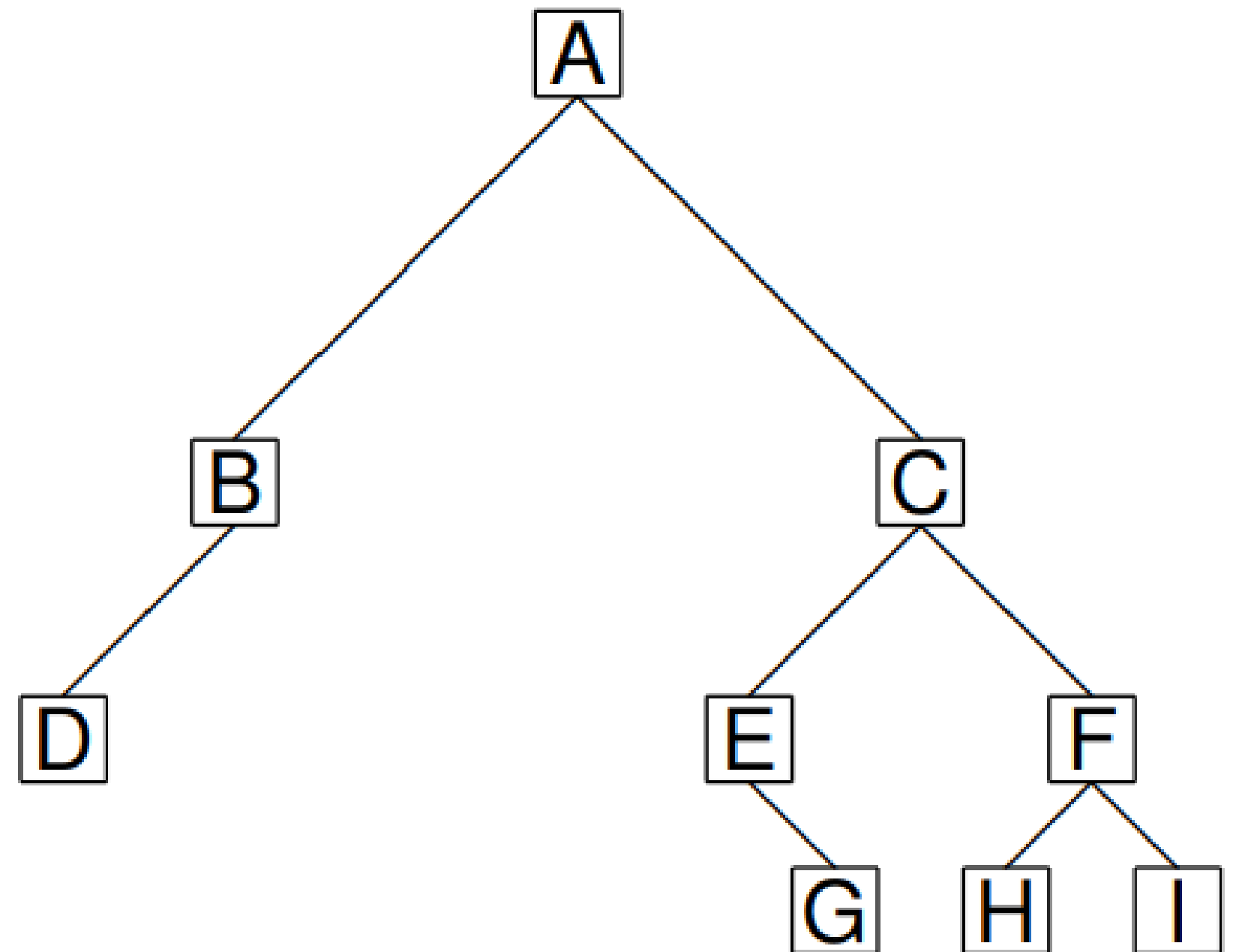
Existem três formas comuns de percurso (ou travessia) em uma árvore binária:

- Pós-ordem (postorder): nesse percurso, a subárvore esquerda é visitada primeiro, seguida pela subárvore direita e, em seguida, pelo nó raiz. Em outras palavras, a ordem de percurso é: esquerda, direita, raiz.

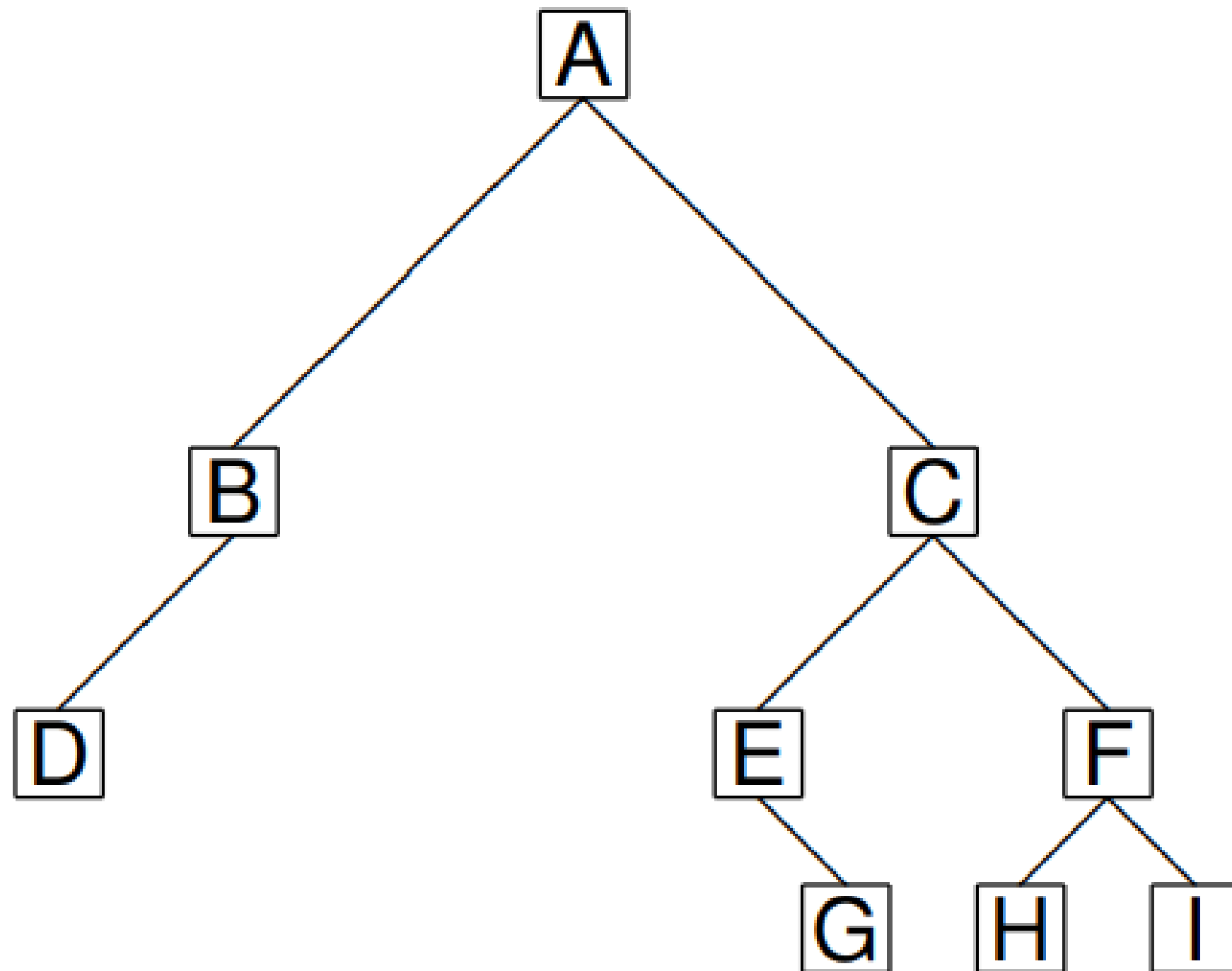
PERCURSO EM UMA ARVORE BINARIA

Percorrer uma árvore binária em pós-ordem:

- 1 Visita a esquerda.
- 2 Visita a direita.
- 3 Vistar a raiz.



PERCURSO EM UMA ARVORE BINARIA



PERCURSO EM POS-ORDEM:

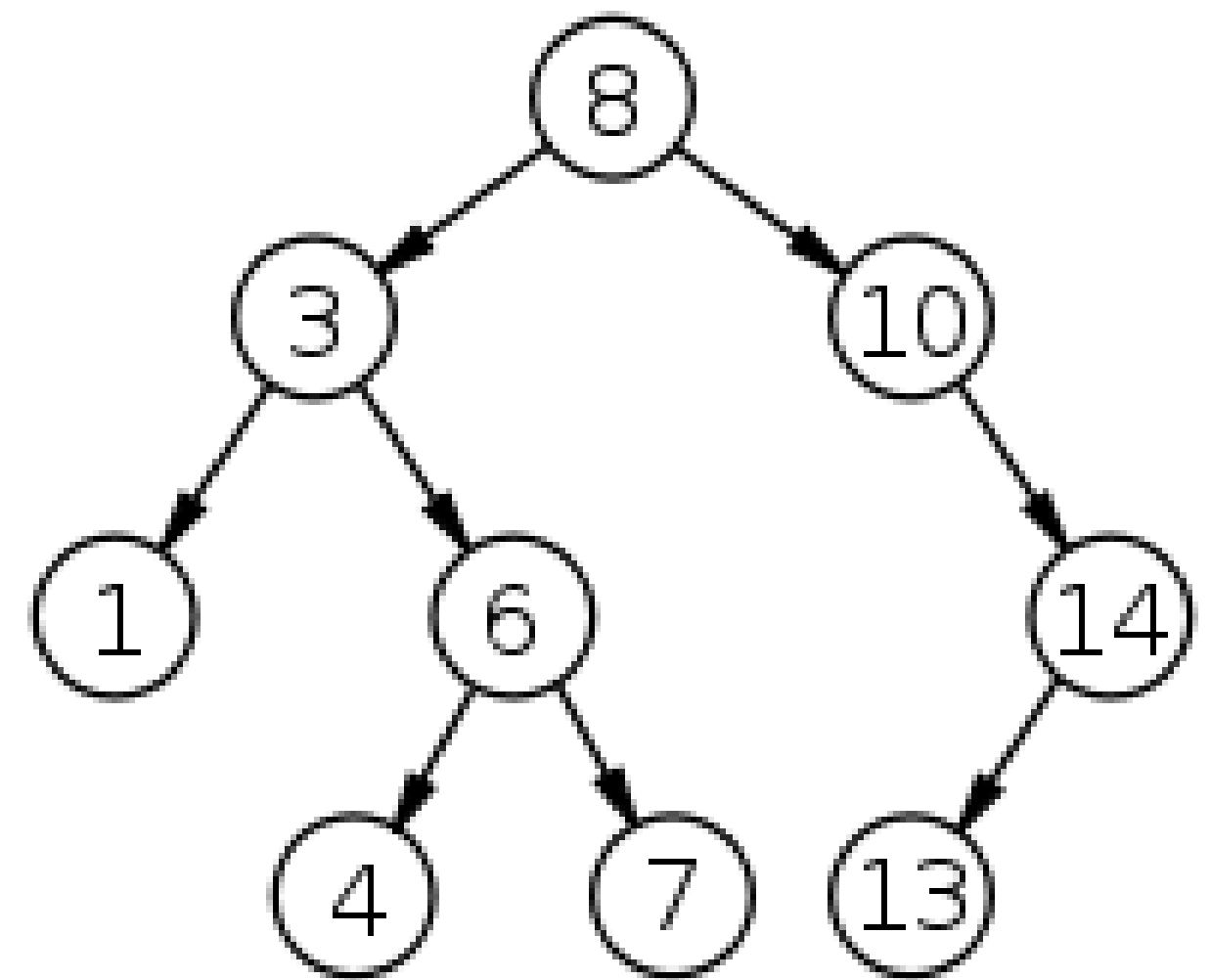
DBGEHIFCA

PERCURSO EM UMA ARVORE BINARIA

Cada forma de percurso tem suas próprias aplicações e pode ser usada para executar diferentes operações na árvore binária. Por exemplo, o percurso em ordem é frequentemente usado para imprimir os elementos da árvore em ordem crescente, enquanto o percurso pré-ordem é útil para criar uma cópia da árvore. O percurso pós-ordem pode ser usado para liberar a memória alocada para a árvore.

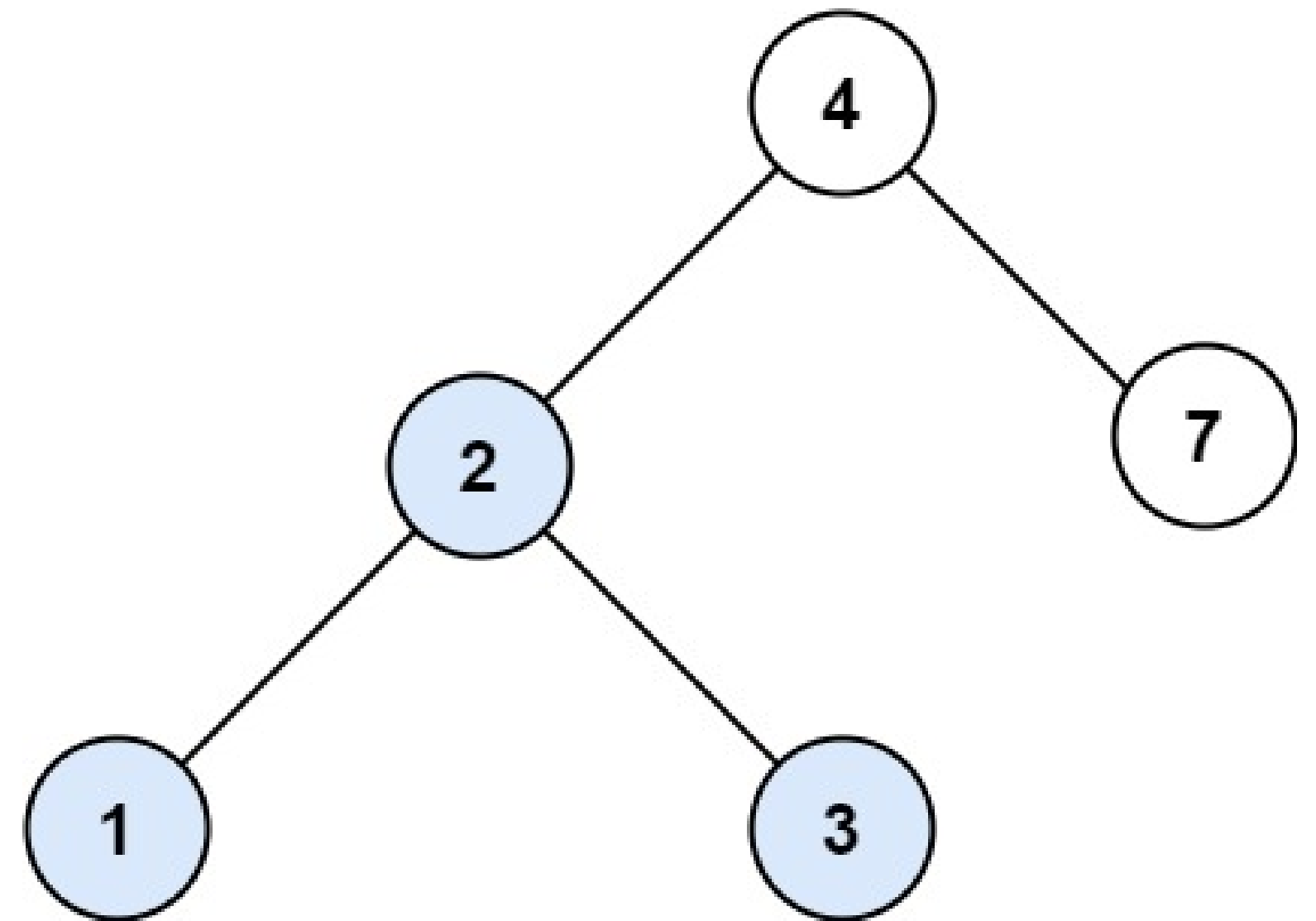
ARVORE BINARIA DE BUSCA

Uma árvore binária de busca (ou binary search tree, em inglês) é uma árvore binária especial em que cada nó contém uma chave de valor e os nós filhos esquerdo e direito contêm valores menores e maiores, respectivamente, do que a chave do nó pai.



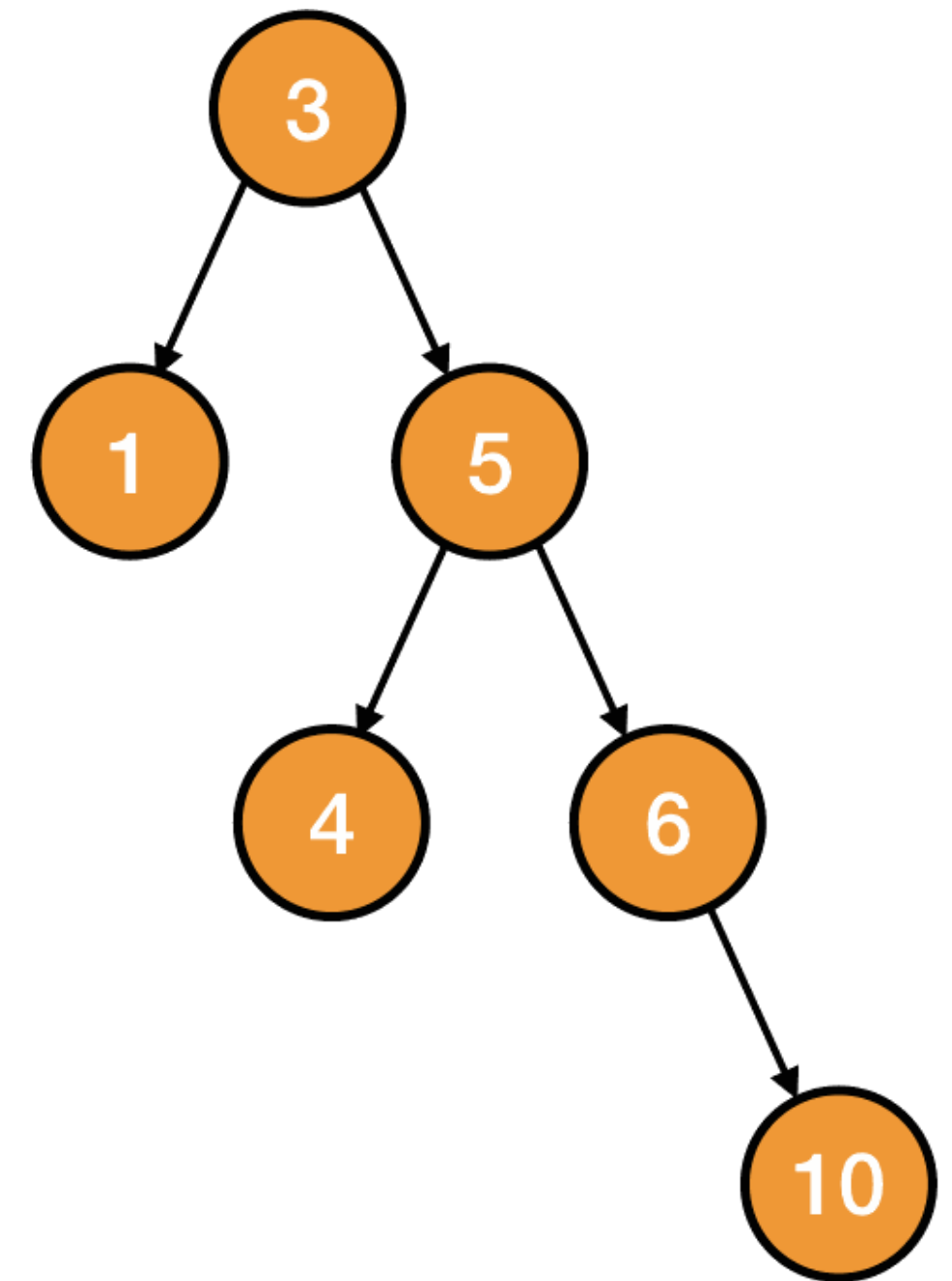
ARVORE BINARIA DE BUSCA

Essa estrutura de árvore permite que as operações de busca, inserção e exclusão sejam executadas com eficiência de tempo médio em $O(\log n)$, onde n é o número de elementos na árvore.



ARVORE BINARIA DE BUSCA

Isso é possível porque a árvore binária de busca possui uma estrutura ordenada que permite a eliminação de metade da árvore em cada iteração da busca, o que a torna uma estrutura de dados muito eficiente para operações de busca.



ALGORITMO DA ABB

```
class No {  
public:  
    int valor;  
    No* esq;  
    No* dir;  
  
    No(int valor) {  
        this->valor = valor;  
        this->esq = nullptr;  
        this->dir = nullptr;  
    }  
};
```

**Estrutura dos
nós**

ALGORITMO DA ABB

```
class ArvBinariaBusca{
public:
    No* raiz;

    ArvBinariaBusca() {
        raiz = nullptr;
    }

    void inserirNo(int valor) {
        No* novoNo = new No(valor);
        if (raiz == nullptr) {
            raiz = novoNo;
            return;
        }
    }
}
```

**Estrutura
inserção de
nós (raiz)**

ALGORITMO DA ABB

```
No* atual = raiz;
while (true) {
    if (valor < atual->valor) {
        if (atual->esq == nullptr) {
            atual->esq = novoNo;
            return;
        }
        else {
            atual = atual->esq;
        }
    }
    else if (valor > atual->valor) {
        if (atual->dir == nullptr) {
            atual->dir = novoNo;
            return;
        }
        else {
            atual = atual->dir;
        }
    }
}
```

**Estrutura
inserção de
nós (esquerda
e direita)**

ALGORITMO DA ABB

```
No* atual = raiz;
while (true) {
    if (valor < atual->valor) {
        if (atual->esq == nullptr) {
            atual->esq = novoNo;
            return;
        }
        else {
            atual = atual->esq;
        }
    }
    else if (valor > atual->valor) {
        if (atual->dir == nullptr) {
            atual->dir = novoNo;
            return;
        }
        else {
            atual = atual->dir;
        }
    }
}
```

**Estrutura
inserção de
nós (esquerda
e direita)**

ALGORITMO DA ABB

```
bool procurar(int valor) {  
    No* atual = raiz;  
  
    while (atual != nullptr) {  
        if (atual->valor == valor) {  
            return true;  
        }  
        else if (valor < atual->valor) {  
            atual = atual->esq;  
        }  
        else {  
            atual = atual->dir;  
        }  
    }  
  
    return false;  
}  
};
```

**Estrutura de
busca de
valores na
ABB**

QUESTÃO 1 - BANCA UFES

O tempo necessário de pesquisa em uma árvore de busca binária varia de acordo com a estrutura dessa árvore. Em árvores de busca binária, o intervalo de variação de tempo de busca é entre

- ☐ A $O(n)$ e $O(n^2)$
- ☐ B $O(n \log n)$ e $O(n^2)$
- ☐ C $O(\log n)$ e $O(n)$
- ☐ D $O(1)$ e $O(\log n)$
- ☐ E $O(1)$ e $O(n \log n)$

DÚVIDAS???

