ESTRUTURAS DE DADOS

Árvores (Conceitos)

Roteiro

- Conceitos Básicos
- Árvore Binária de Busca
  - Busca
  - Inserção
  - Remoção
- Percursos

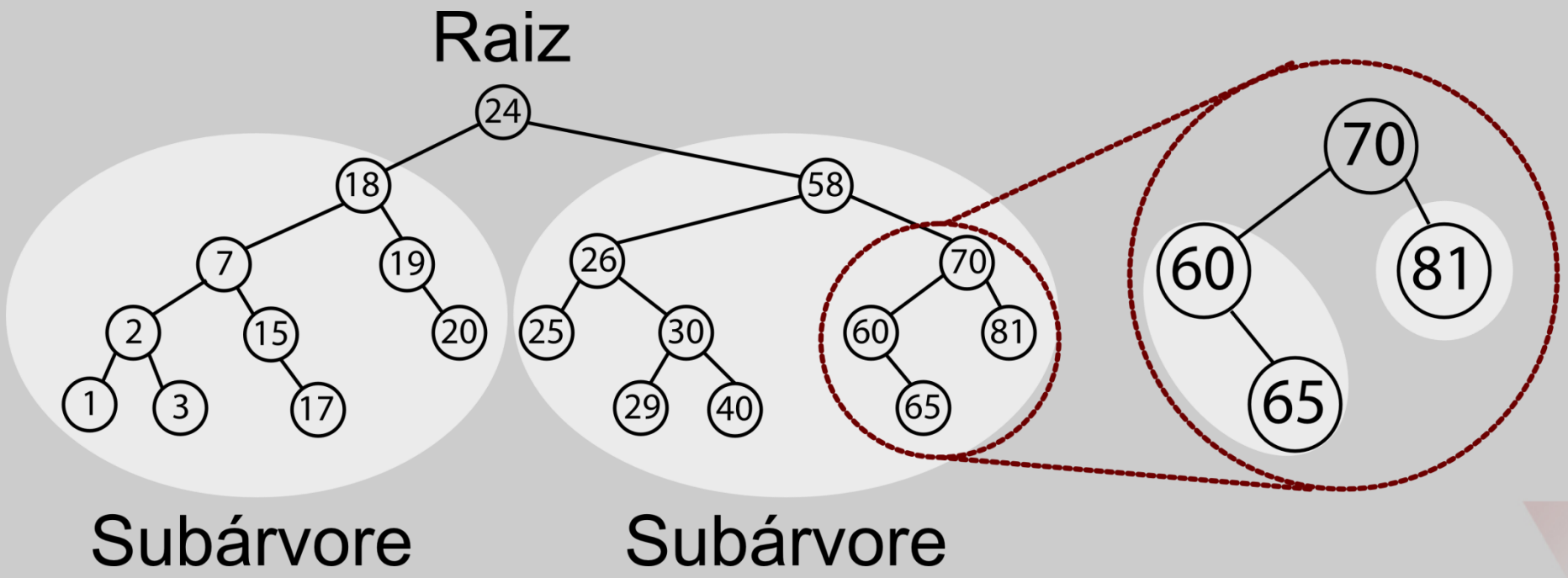
Conceitos Básicos

Uma árvore é um conjunto de nós em que existe um nó raiz *r*, que contém zero ou mais subárvores cujas raízes são ligadas diretamente a *r.*

- **Uma subárvore é também uma árvore.**

Uma árvore não é uma estrutura linear, não há um sucessor e um predecessor por nó.

- **Estruturas lineares não são adequadas para representar hierarquia nos dados.**

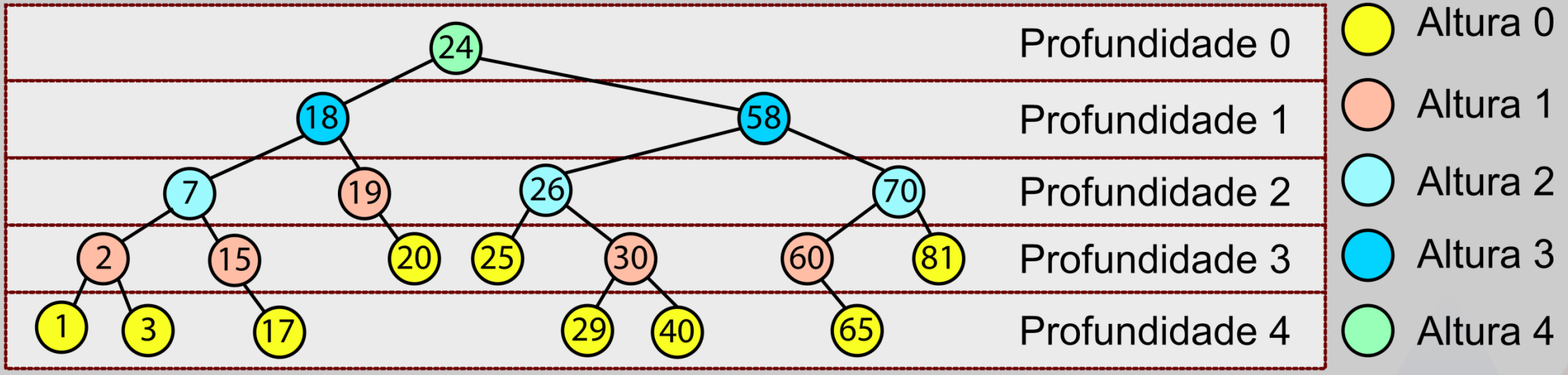


Grau de um nó: número de subárvores desse nó.

Nó folha: nó de grau 0 (zero).

Nó interno: nó de grau maior que 0 (zero).

Descendentes: nós abaixo de um determinado nó



Altura de um nó: comprimento do caminho mais longo entre o nó até uma folha.

Altura da árvore: altura do nó raiz.

Profundidade de um nó: distância percorrida da raiz até o nó.

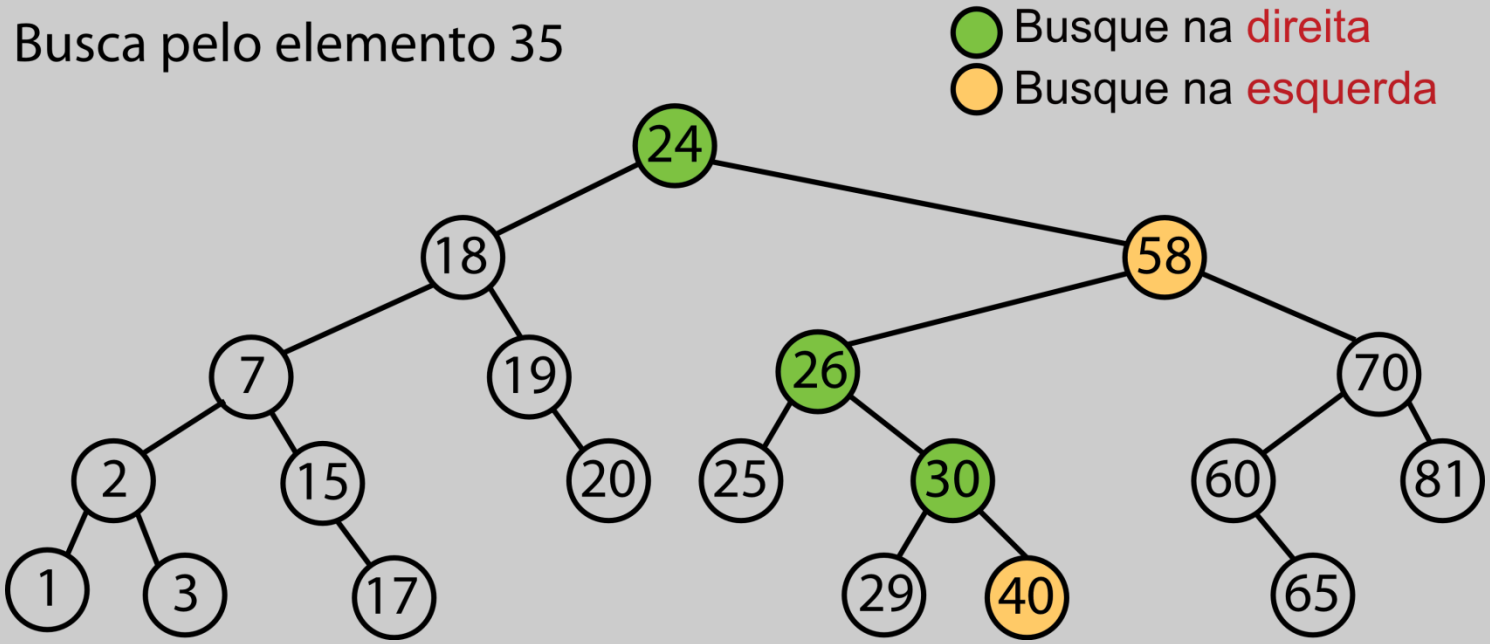
Árvore Binária: árvore em que abaixo de cada nó existem no máximo duas subárvores.

Árvore Binária de Busca

Árvore binária em que, a cada nó, todos os registros com chaves menores que a deste nó estão na subárvore da esquerda, enquanto que os registros com chaves maiores estão na subárvore da direita.

Inserções, remoções e buscas possuem número de comparações proporcional à altura da árvore.

Buscas em Árvores Binárias de Busca

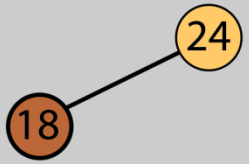


Se chave igual a nó, então achamos. Se chave maior que nó, pesquisamos na subárvore da direita. Caso contrário, na da esquerda. Se alcançarmos um nó nulo, então paramos.

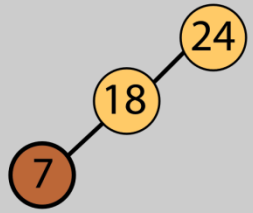
Inserção em Árvores Binárias de Busca

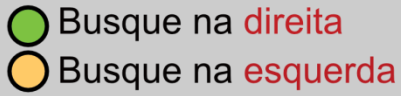
Inserir: 24

Inserir: 18

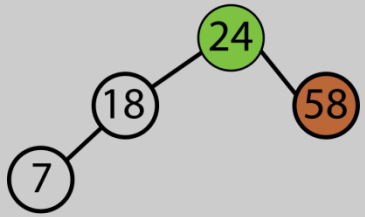


Inserir: 07

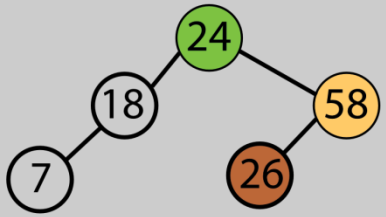




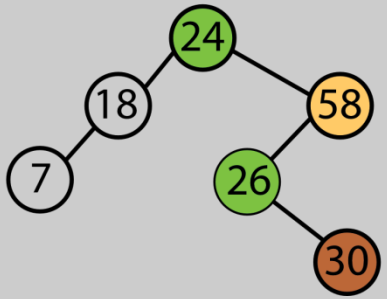
Inserir: 58



Inserir: 26



Inserir: 30



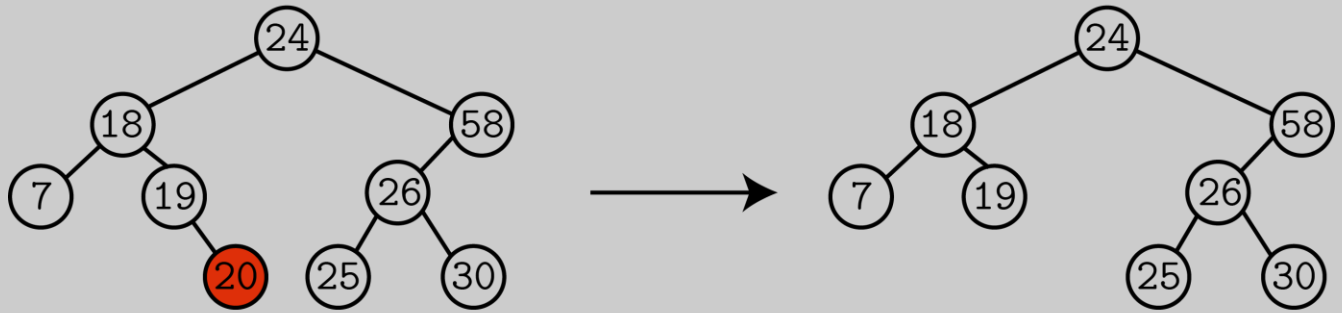
Supondo que não permitimos duplicação na árvore, inserimos apenas se o elemento não existe. Nesse caso, basta inserir o elemento na posição que ele estaria se fosse buscado.

Remoção em Árvores Binárias de Busca

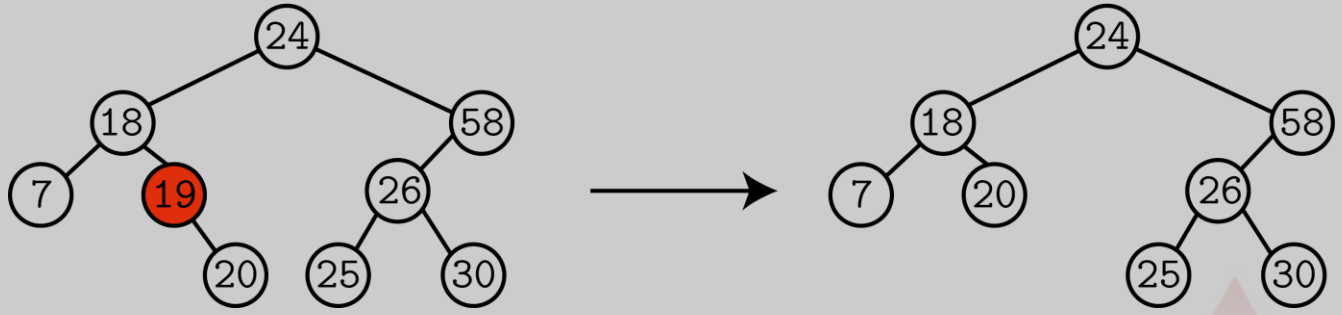
Se o nó a ser removido não possui filhos, simplesmente remova.

Se o nó possui um filho, então remova e coloque o filho no lugar.

Remover 20



Remover 19

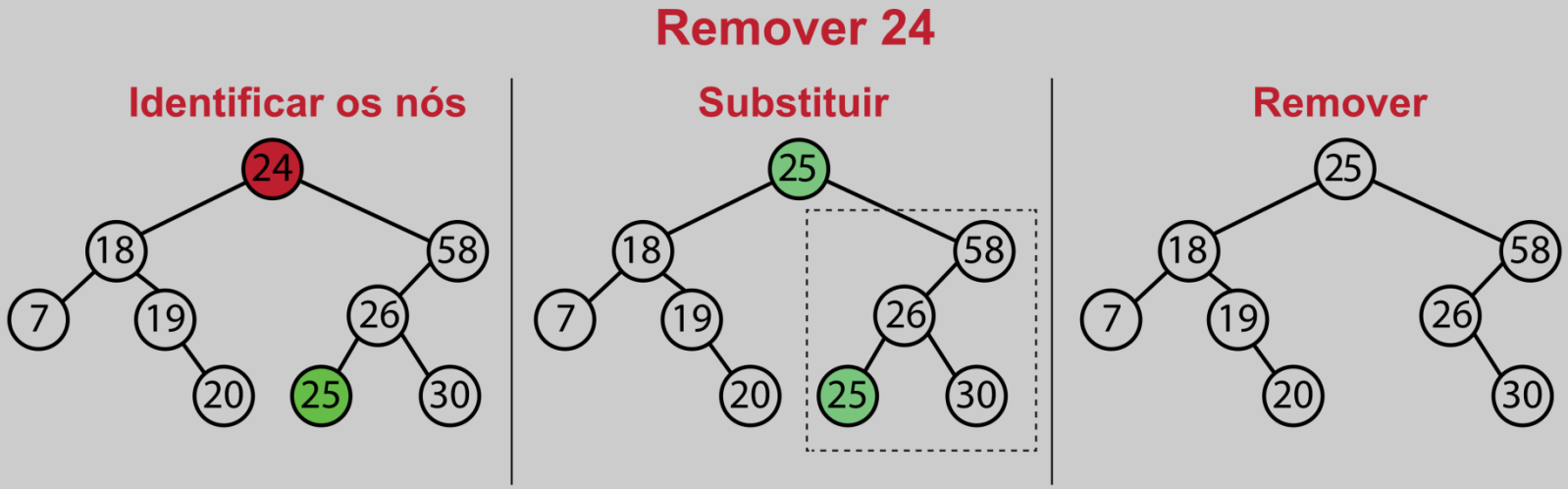


Remoção em Árvores Binárias de Busca

Se o nó possui mais de um filho?

Opção 1:

Se o nó possui mais de um filho, então substitua pelo sucessor lógico antes de remover.



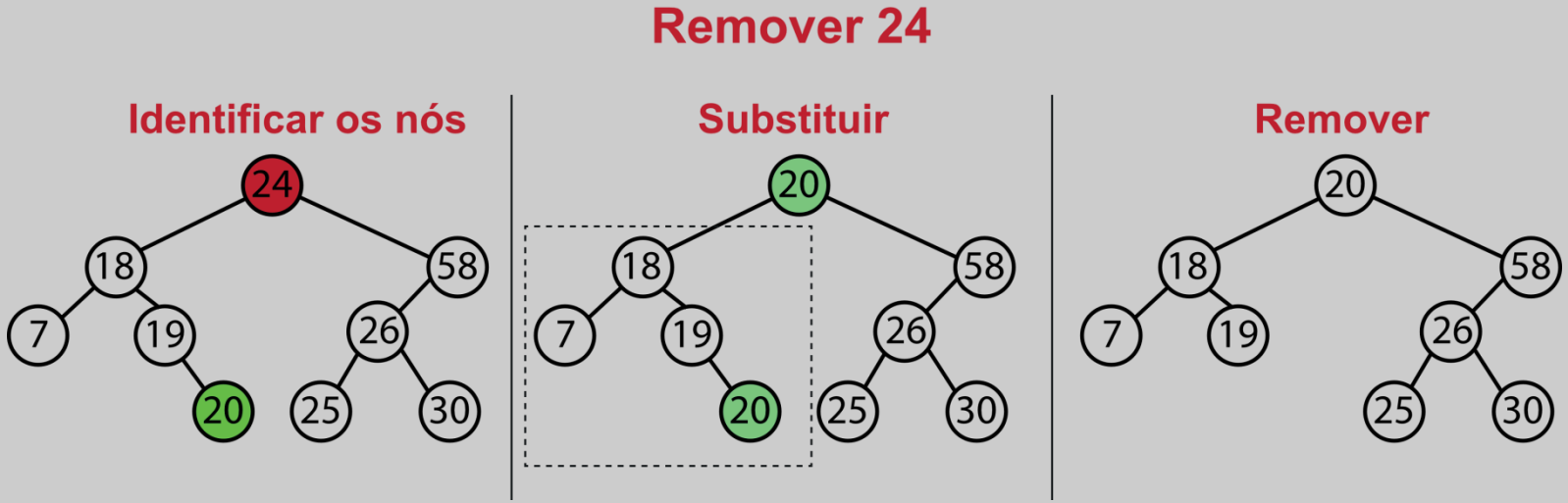
O sucessor lógico é sempre o elemento mais à esquerda na subárvore da direita.

Remoção em Árvores Binárias de Busca

Se o nó possui mais de um filho?

Opção 2:

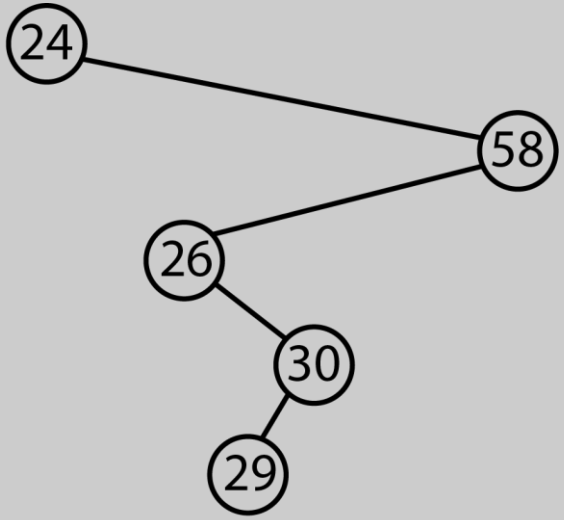
Se o nó possui mais de um filho, então substitua pelo antecessor (ou predecessor) lógico antes de remover.



O predecessor lógico é sempre o elemento mais à direita na subárvore da esquerda.

Nos algoritmos de busca, inserção e remoção, no pior caso, o número de comparações é proporcional à altura da árvore.

- **As buscas são eficientes em árvores balanceadas.**
- **Em árvores degeneradas, as operações deixam de ser eficientes.**



Número de Elementos: 5
Número Máximo de Comparações: 5

Em uma próxima aula, veremos como garantir que uma árvore binária de busca se mantenha balanceada.

Percursos

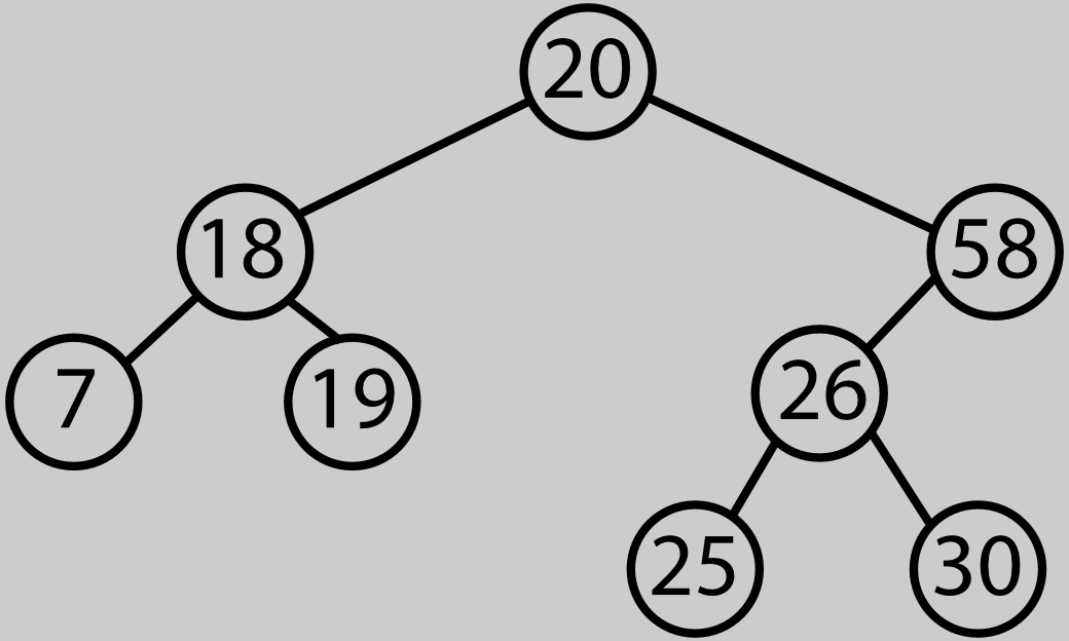
Em muitos algoritmos, precisamos percorrer os nós de maneira sistemática, visitando cada nó apenas uma vez.

Existem três tipos de percursos mais comuns em árvores binárias:

- **Pré-ordem**
- **Pós-ordem**
- **In-ordem**

A diferença entre os caminhamentos se refere ao momento em que visitamos o nó central.

- **Sempre visitamos a subárvore da esquerda antes de visitarmos a subárvore da direita.**
- **Pré-ordem: visitamos, a partir da raiz, primeiramente o nó raiz, depois os nós da esquerda, depois os da direita.**
- **Pós-ordem: visitamos, a partir da raiz, primeiramente os nó da esquerda, depois os nós da direita e depois concluímos visitando o nó raiz.**
- **In-ordem: visitamos, a partir da raiz, primeiramente os nós da esquerda, depois visitamos o nó raiz, e depois concluímos visitando os nós da direita.**



Pré-ordem: 20, 18, 7, 19, 58, 26, 25, 30

Pós-ordem: 7, 19, 18, 25, 30, 26, 58, 20

In-ordem: 7, 18, 19, 20, 25, 26, 30, 58

ESTRUTURAS DE DADOS

Árvores (Conceitos)