Árvores Binárias- Árvore de Busca Binária

Códigos de Alta Performance

PROFa. PATRÍCIA MAGNA - profpatricia.magna@fiap.com.br

Uma Árvore Binária é uma árvore que obedece as seguintes condições:

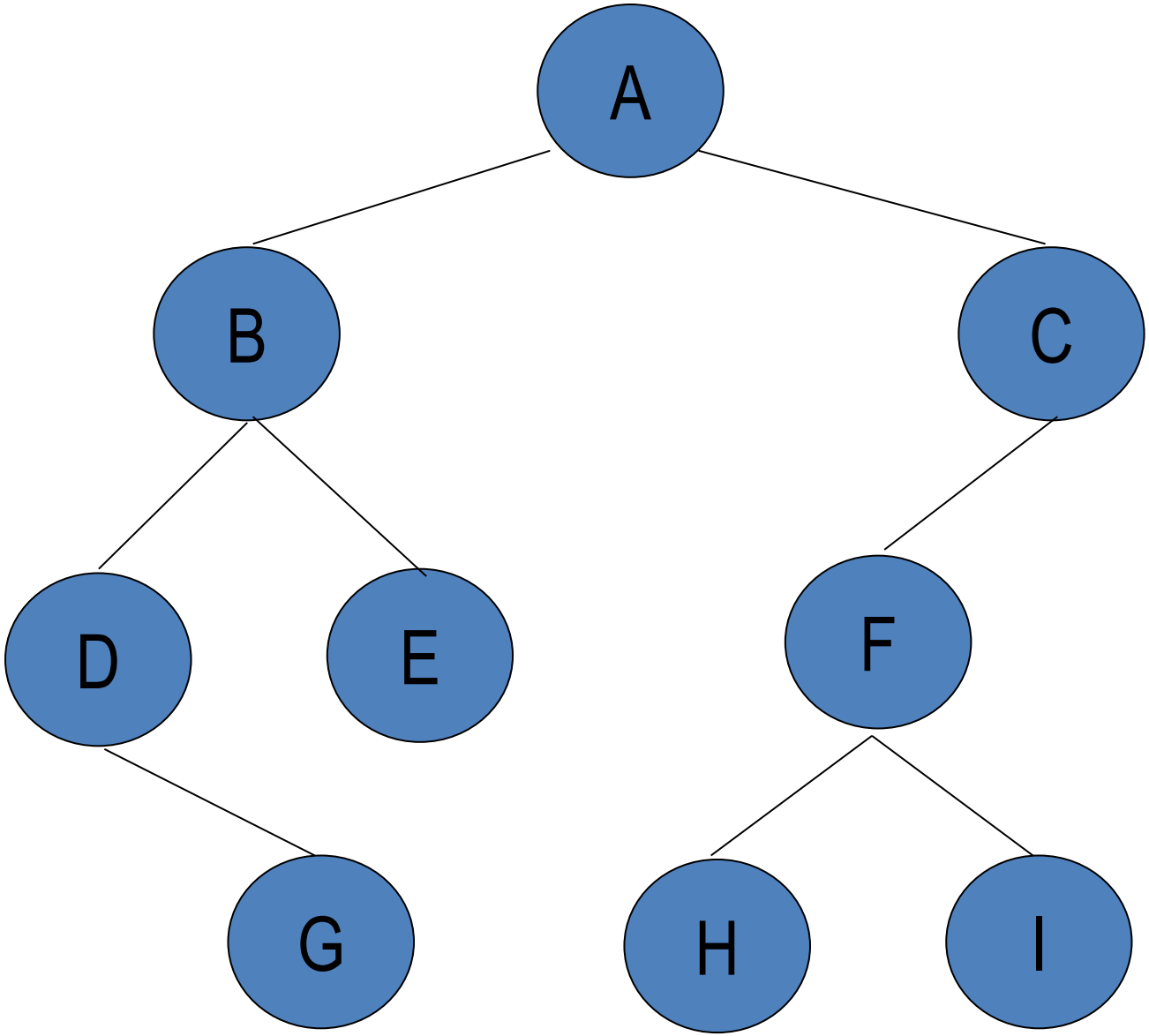
Conjunto finito de elementos que está vazio

OU

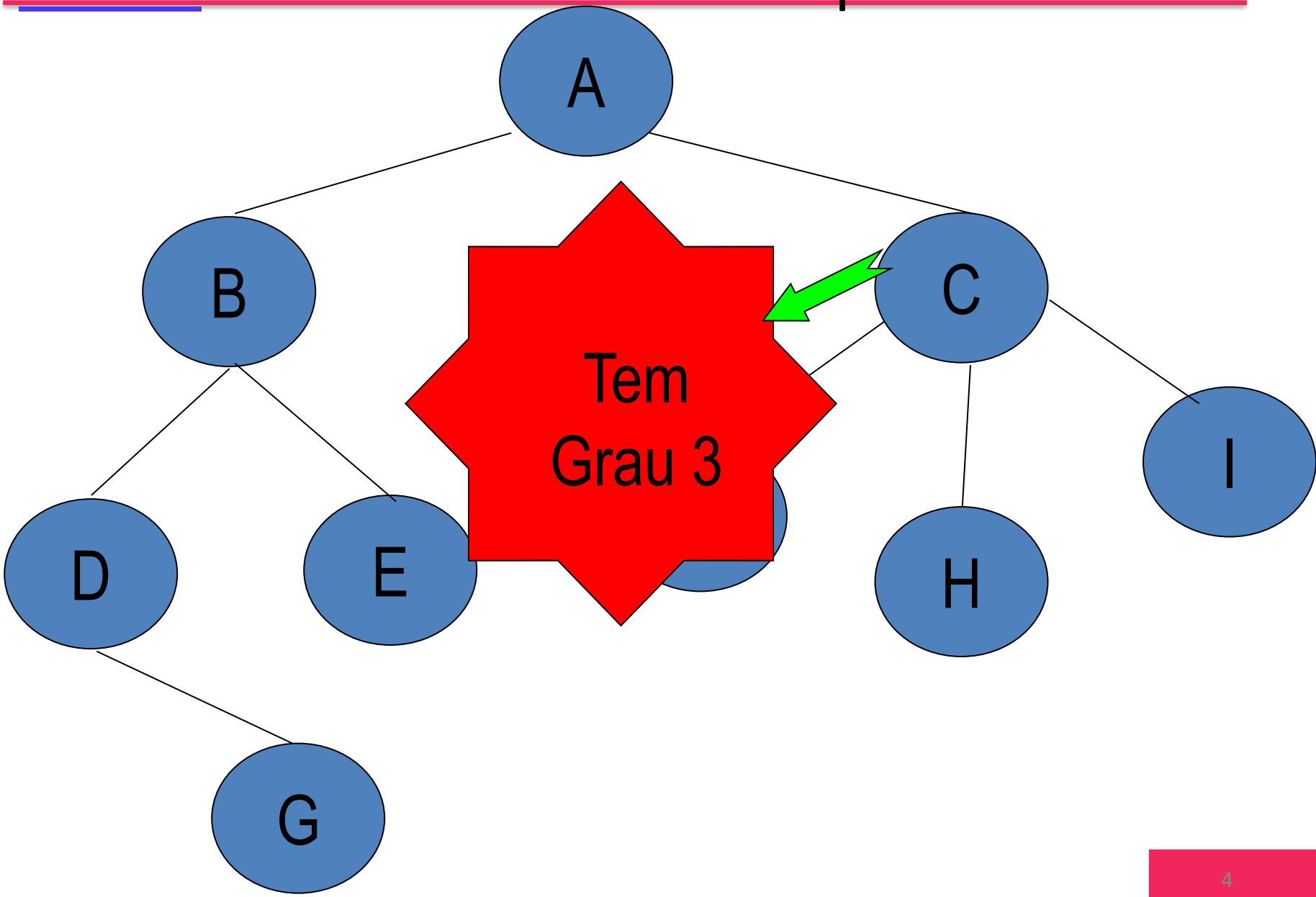
Conjunto finito de elementos particionados (divididos) em 3 subconjuntos disjuntos:

1º: contém um único elemento: RAIZ.
2º: sub-árvore direita (pode estar vazia)
3º: sub-árvore esquerda (pode estar vazia)

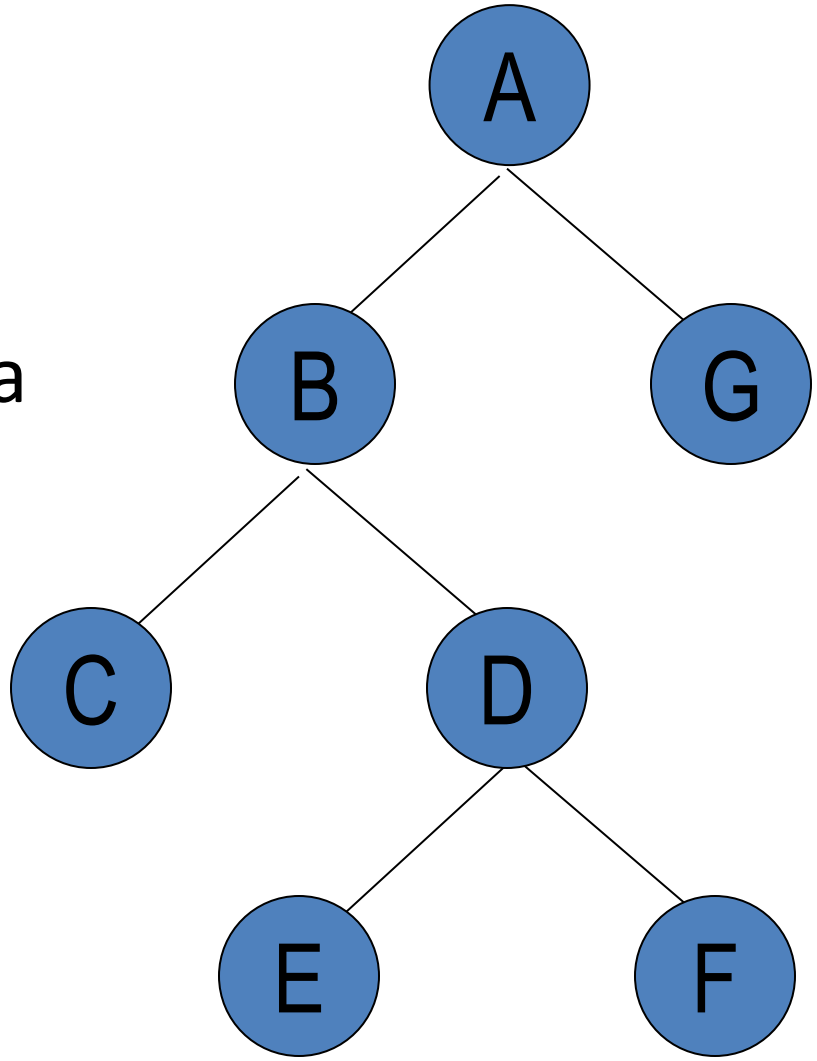
Árvore Binária – Exemplo 5



NÃO É Árvore Binária – Exemplo 6

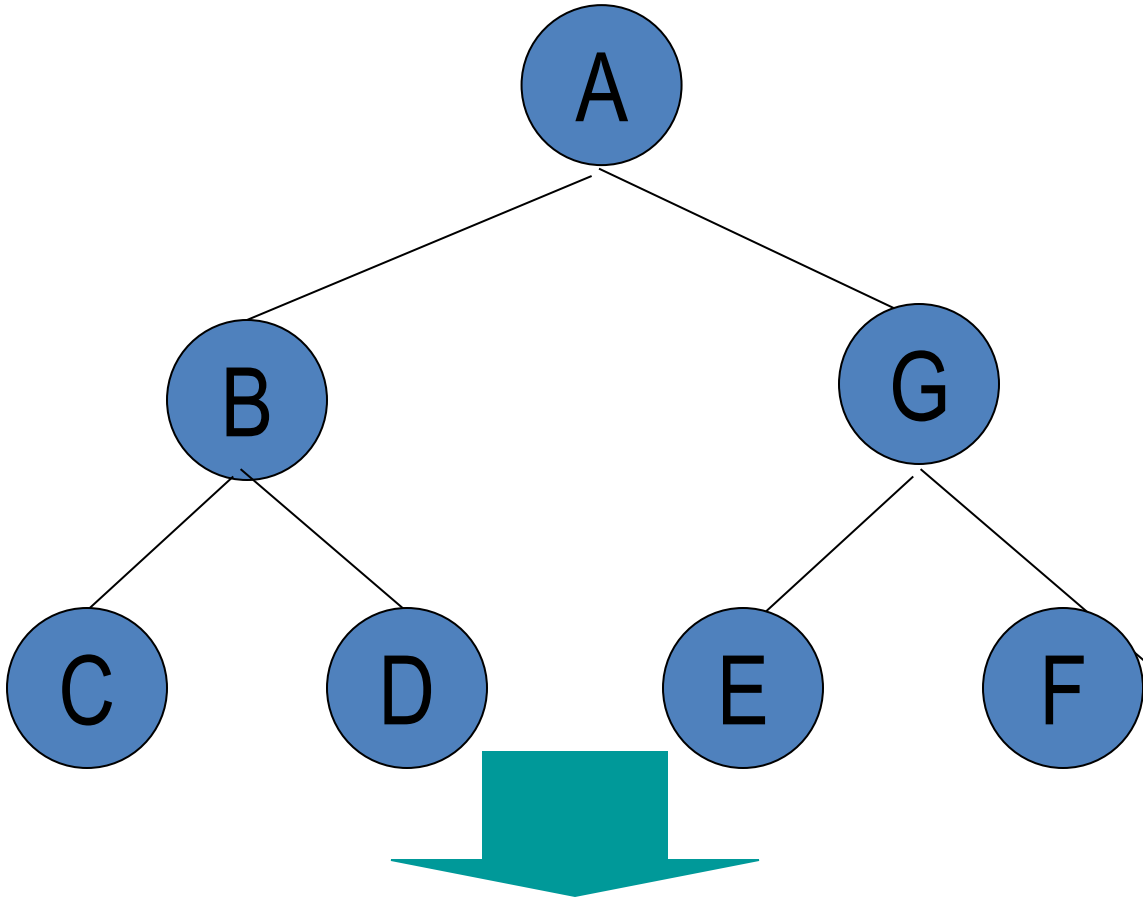


Árvore Estritamente Binária: se todo nó que não é folha tiver sub-árvores esquerda e direita não vazias.



Árvore Binária Completa de Profundidade *d*:

é a árvore estritamente binária em que todas as folhas estejam no nível *d*.



Árvore binária completa de profundidade 3

- O número total (NT) de nós numa árvore binária completa de profundidade d é igual a somatória do número de nós em cada nível entre 0 e *d*:

$$NT = \sum_{j=1}^{d} 2^{j-1}$$

2^j representa a quantidade de nós do nível j:

j = 1 (nível 1) -> NT = 2^0 = 1

j = 2 (nível 2) -> NT = $2^0 + 2^1 = 1+2 = 3$

j = 3 (nível 3) -> NT = $2^0 + 2^1 + 2^2 = 1+2+4 = 7$

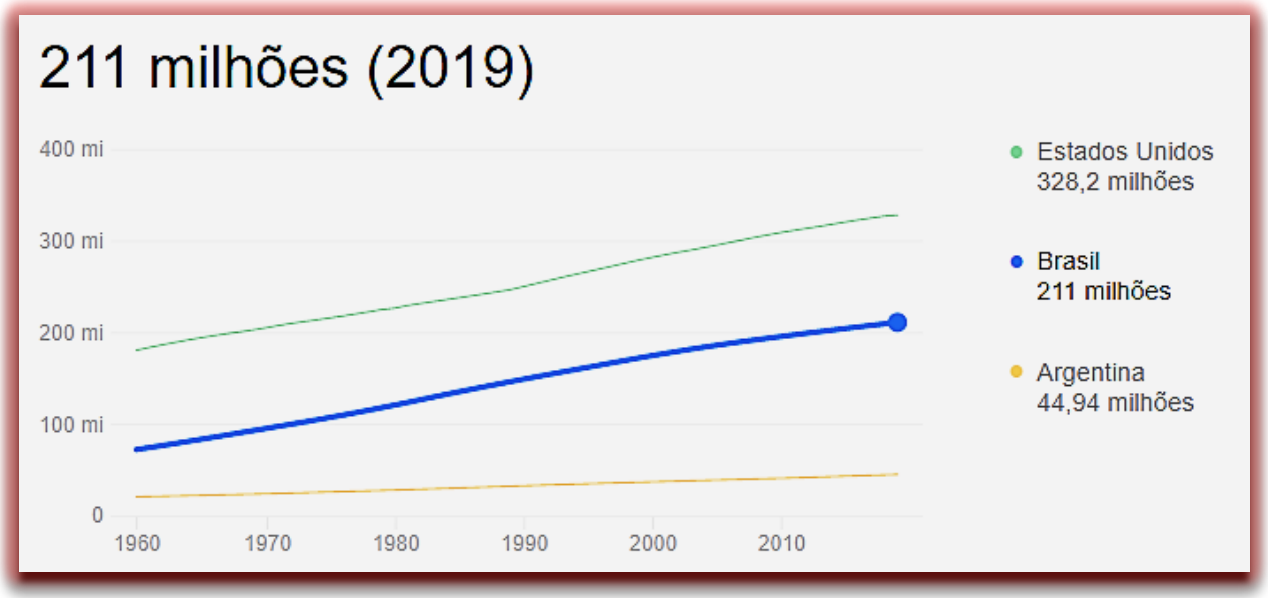
j = 4 (nível 4) -> NT = $2^0 + 2^1 + 2^2 + 2^3 = 1+2+4+8=15$

Árvore Binária Completa

N° nós = $2^{\text{níveis}} - 1$

Níveis	Número de Nós
4	15
5	31
10	1.023
11	2.047
20	1.048.575
21	2.097.151
27	134.217.727
28	268.435.455

Exemplo de Eficiência



Níveis	Número de Nós
27	134.217.727
28	268.435.455

Com 28 comparações podemos encontrar qualquer cidadão pelo RG na base de dados

Árvore Binária Quase Completa

- Cada folha na árvore deve estar no nível d ou no nível d-1

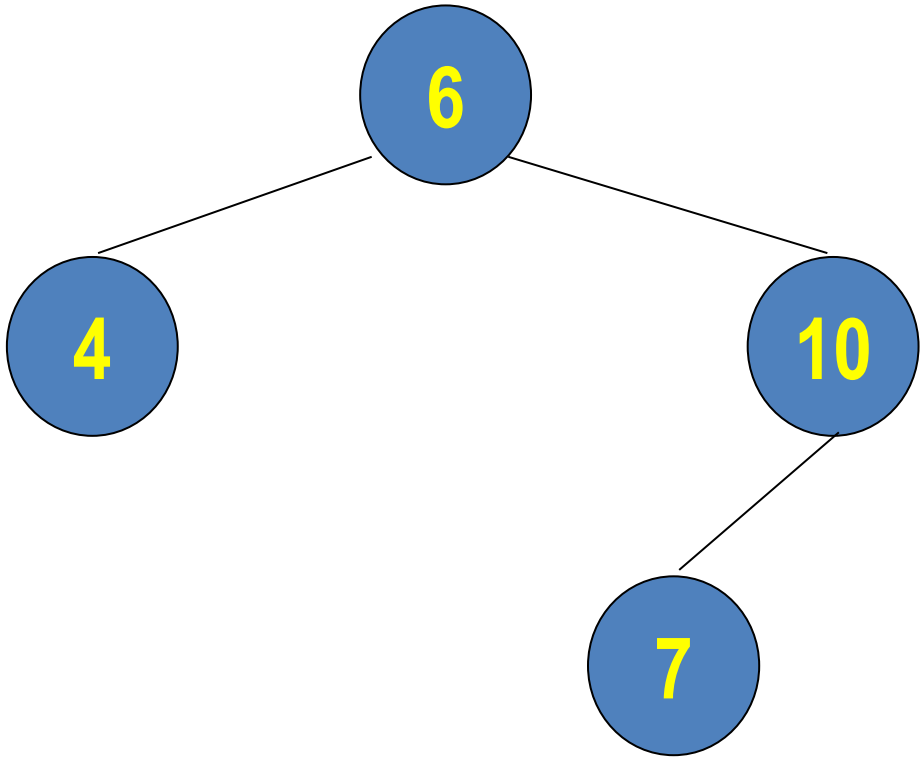
- Para cada nó X na árvore com um descendente direito no nível d, todos os descendentes esquerdos de X que forem folhas devem estar no nível d.

ÁRVORE de BUSCA BINÁRIA

Árvore de Busca Binária - ABB

Propriedades a serem seguidas para formação da ABB:

- O dado em qualquer nó N é maior que todos os dados da sub-árvore esquerda de N
- O dado em N é menor ou igual que todos os dados da sub-árvore direita de N



Exemplo e Exercício

Planilha **ARVORE BINARIA**

Árvores de Busca Binárias - Conceito

FIAP

Uma árvore de busca criada a partir de um conjunto de dados não é única, dependendo da ordem de inserção de dados indicada.

Se for alterada a ordem em que os valores são inseridos nos exercícios feitos na planilha pode-se observar que, inclusive, a ABB não fica com todos os nós folha no mesmo nível.

Árvores de Busca Binária - Conceito

A grande utilidade da árvore binária de busca é armazenar dados, sendo que as operações de inserção, remoção e, em especial, a localização de dados são freqüentemente realizadas.

Árvores Binárias de Busca - Conceito

FIAP

- Uma árvore de binária de busca pode sofrer alterações (inserções e remoções) após ter sido criada.

•Exemplo: Um compilador pode usar uma árvore para armazenar os comandos válidos de uma linguagem de programação.

- •neste caso a árvore não sofre alterações após criada.
- •Também pode manter uma árvore para conter os símbolos definidos no programa em processamento (nomes de variáveis).

Árvores Binárias - Percursos

- Não existe uma ordem “natural” para os nós de uma árvore.
- São usados diferentes ordenamentos de percursos em diferentes casos.
- Definiremos 3 métodos de percurso.
- Todos eles envolvem visitar a raiz e percorrer suas sub-árvores esquerda e direita recursivamente. A diferença entre eles é a ordem em que essas operações são realizadas.
- Os métodos são usados para a árvore não vazia.

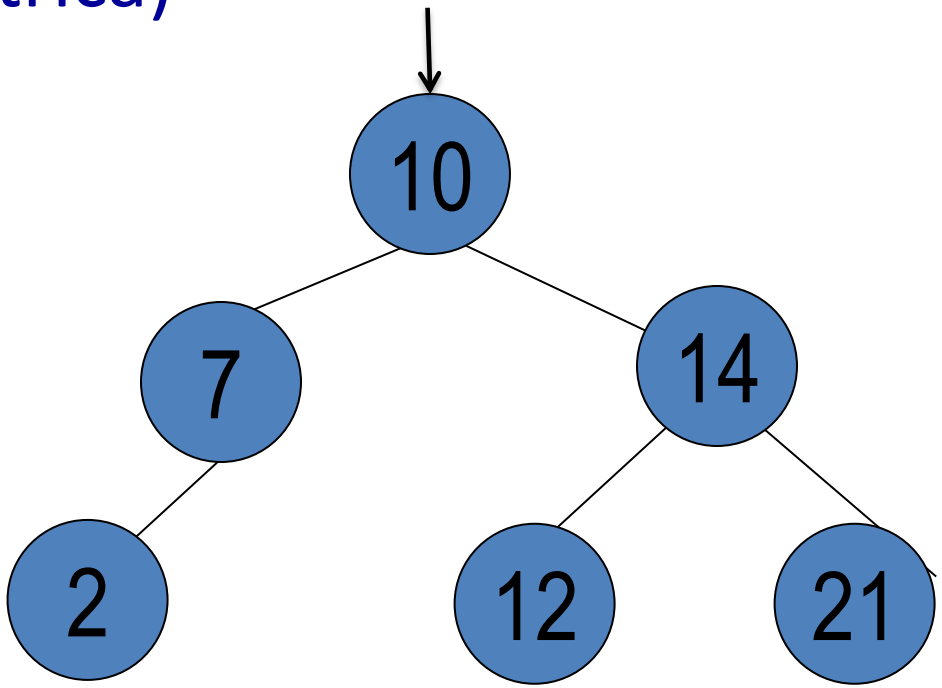
- EM Ordem (Ordem Simétrica)

RECURSIVO

1 – Percorremos a subárvore esquerda em ordem simétrica.

2 – Apresentamos DADO

3 – Percorremos a subárvore direita em ordem simétrica.



Em Ordem: 2 7 10 12 14 21

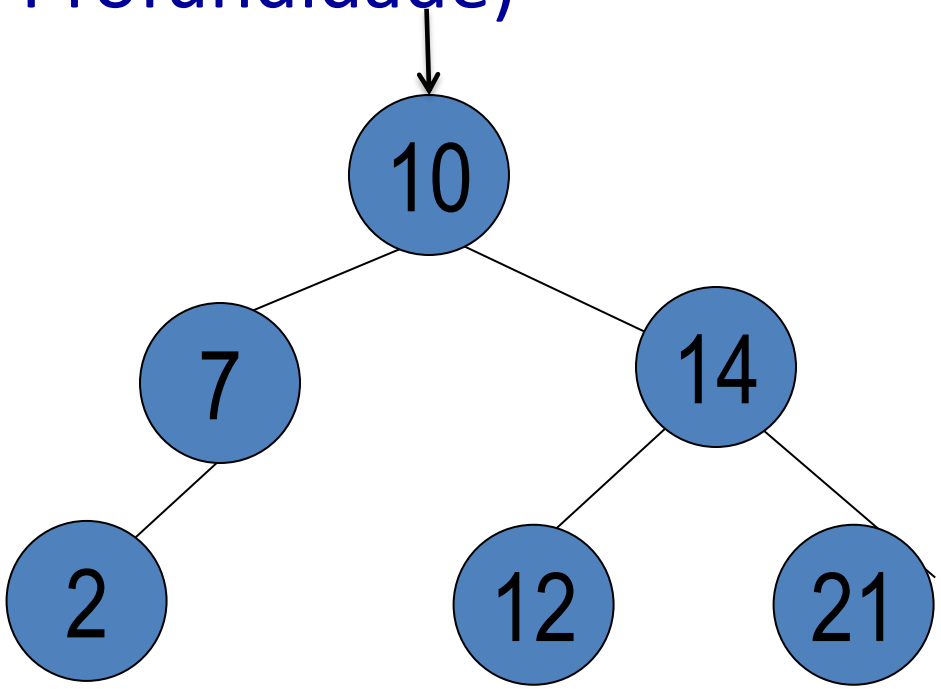
- Pré-Ordem (Percurso em Profundidade)

RECURSIVO

1 – Apresentamos DADO

2 – Percorremos a subárvore esquerda em ordem prévia.

3 – Percorremos a subárvore direita em ordem prévia.



Pré – Ordem: 10 7 2 14 12 21

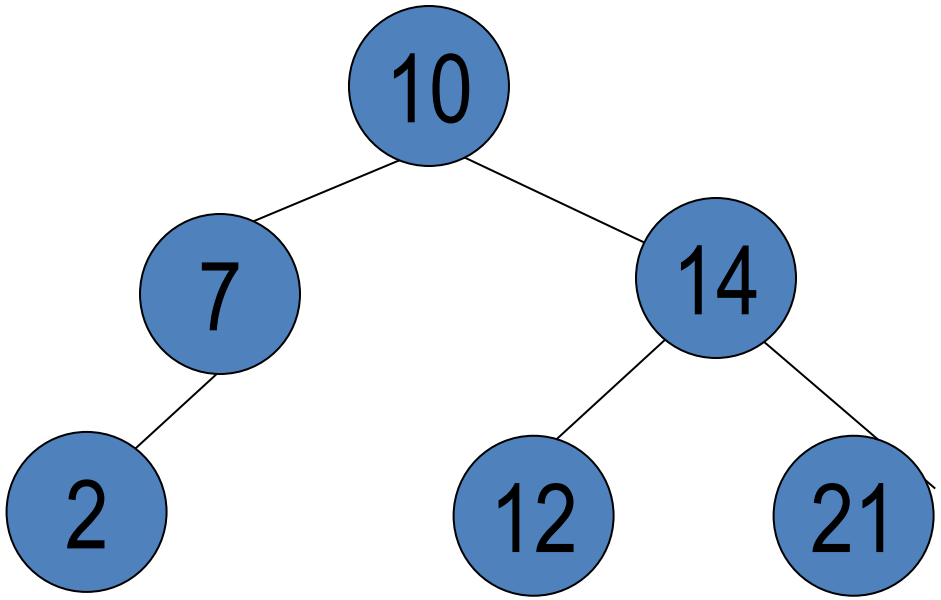
- Pós-Ordem

RECURSIVO

1 – Percorremos a subárvore esquerda em ordem posterior.

2 – Percorremos a subárvore direita em ordem posterior.

3 – Apresentamos DADO



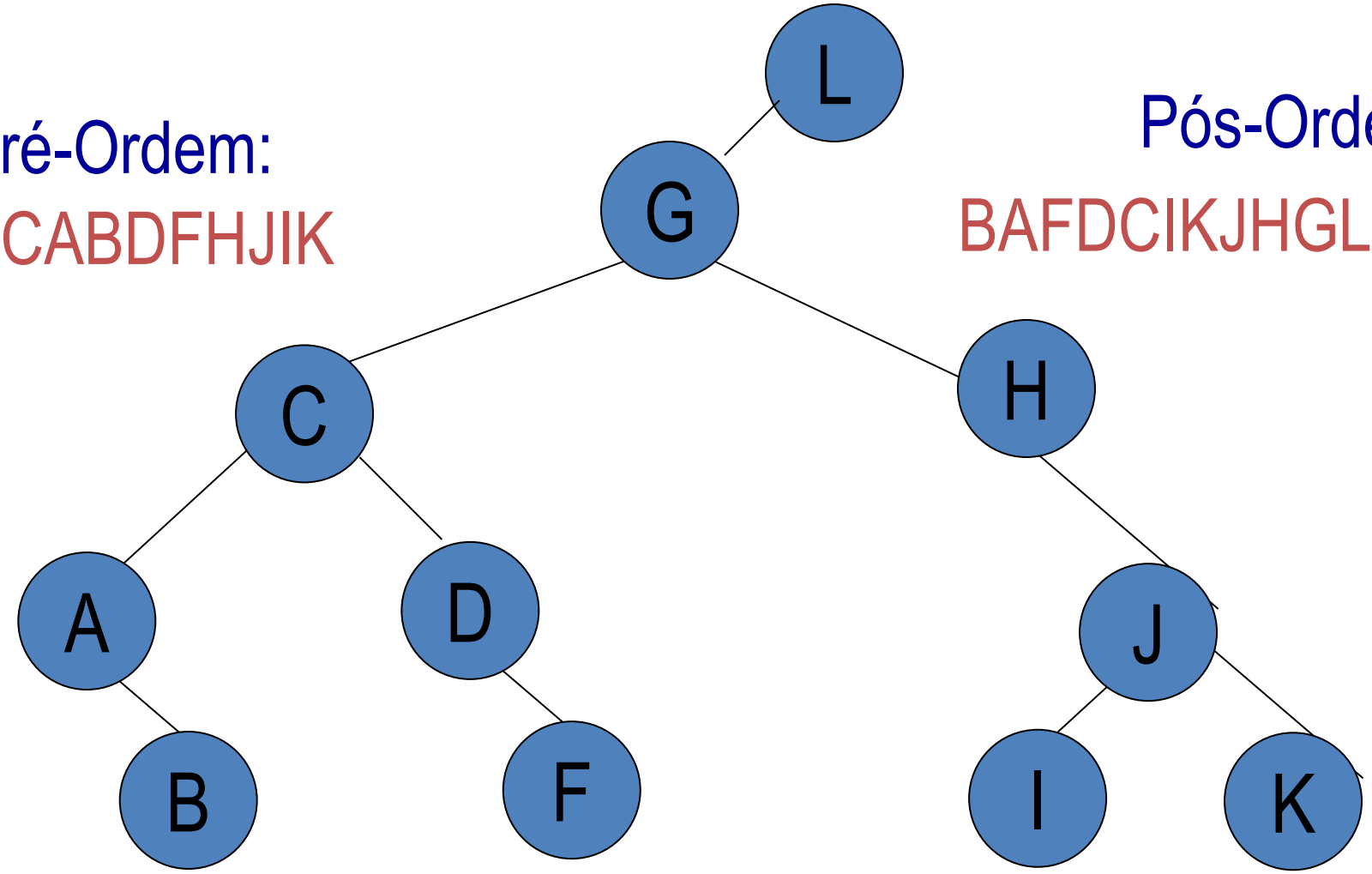
Pós-Ordem: 2 7 12 21 14 10

Árvores Binárias – Exercício: Percurso

FIAP

Pré-Ordem:
LGCABDFHJIK

Pós-Ordem:
BAFDCIKJHGL



Em Ordem: ABCDFGHIJKL

REFERÊNCIAS

- TENENBAUM, A.M. E outros - **Estruturas de Dados usando C**. Makron Books do Brasil Editora Ltda, SP.
- PEREIRA, S. L. **Estrutura de Dados Fundamentais.** São Paulo: Érica.
- FORBELLONE, A.L.V. & EBERSPÄCHER, H.F. – **Lógica de Programação: A Construção de Algoritmos e Estruturas de Dados.** Makron Books, São Paulo, SP
- ASCENCIO,A.F.G e ARAÚJO, G.S. – Estruturas de Dados: Algoritmos, Análise da Complexidade e Implementação em JAVA e C/C++