

Estrutura de Dados 2

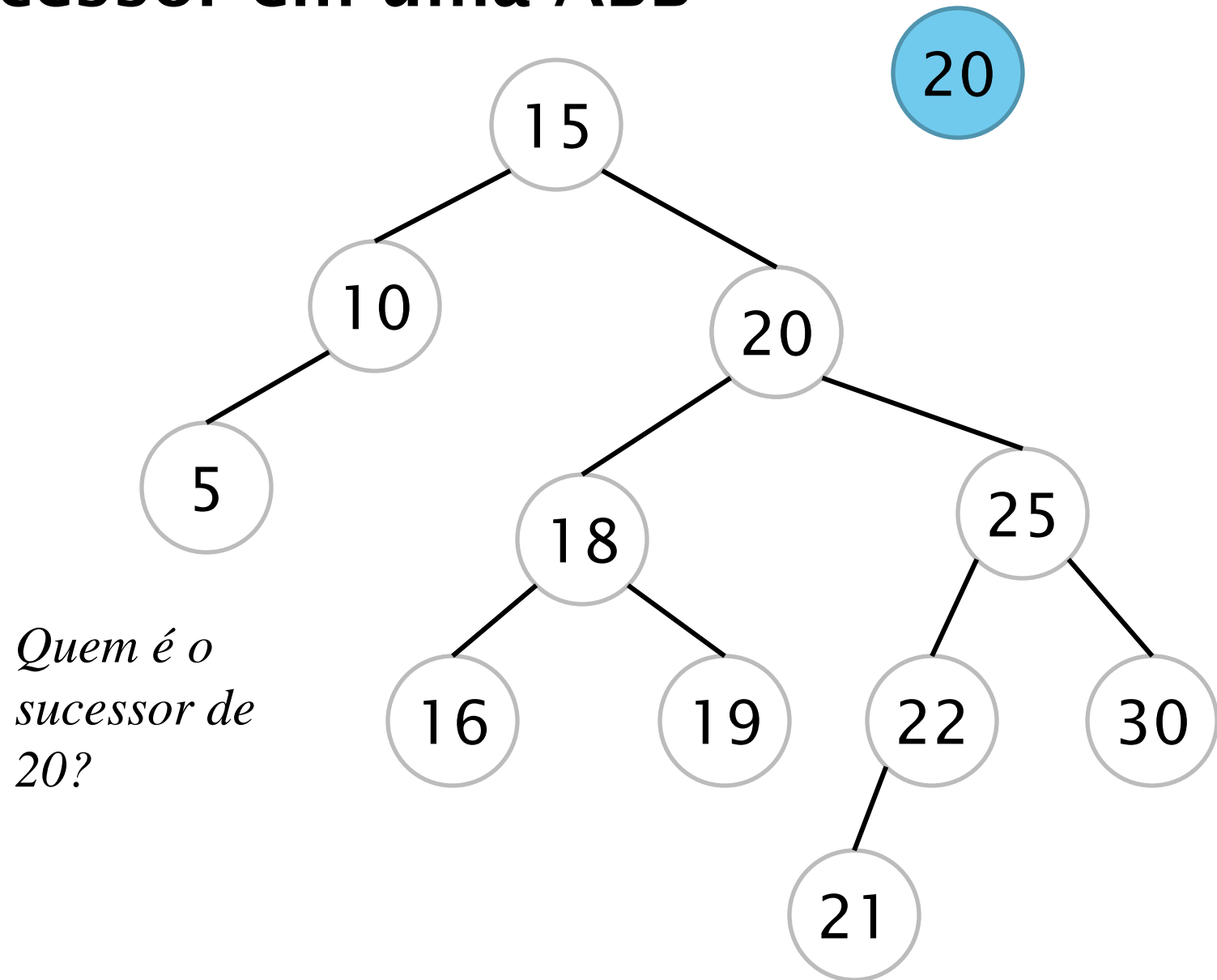
Árvore Binária de Busca (ABB) - Navegação

Prof. Rafael Oliveira Vasconcelos

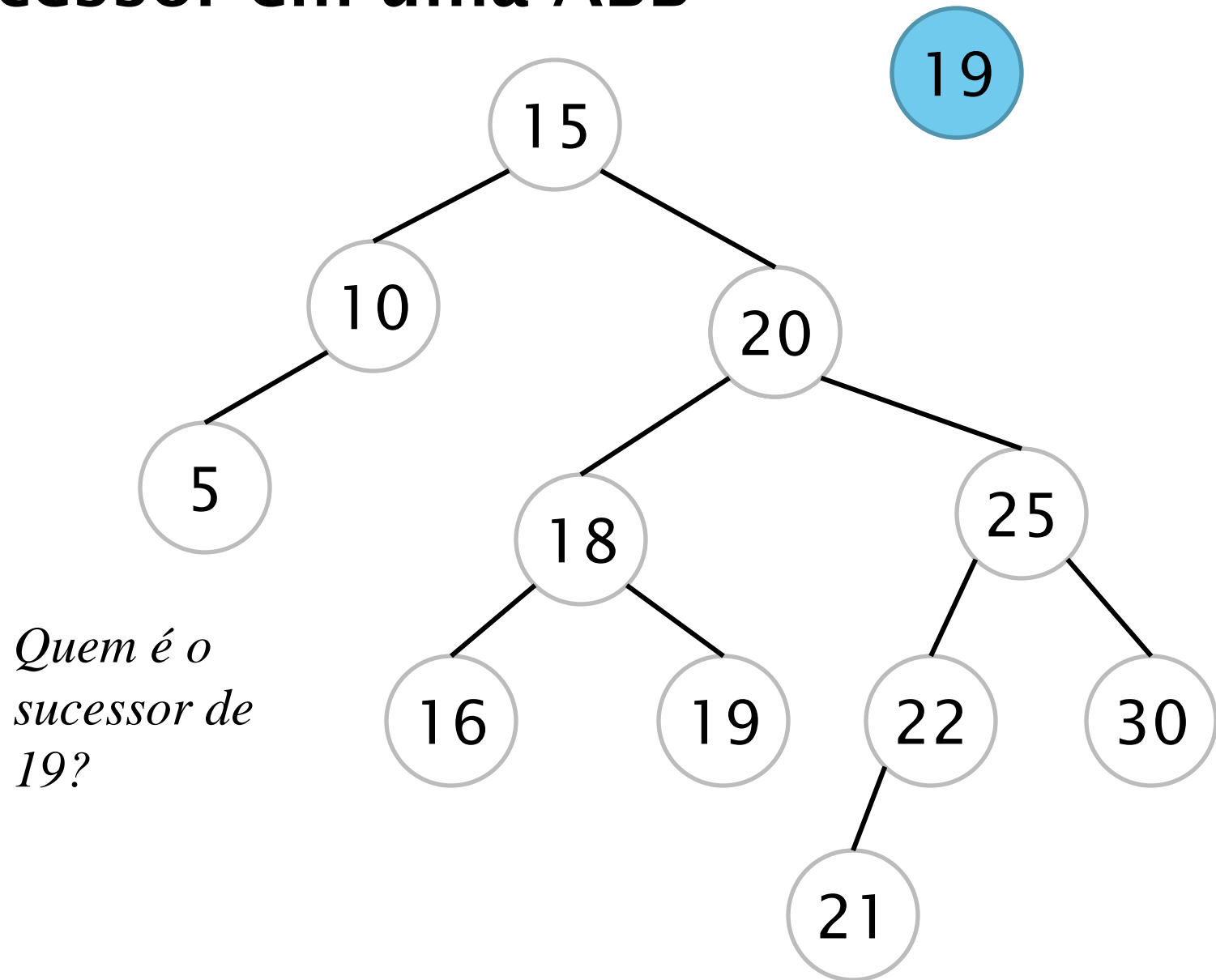
Menor nó em uma ABB

- É o nó mais à esquerda da árvore
1. Começando pelo nó raiz
 2. Se a árvore for vazia retorne NULL
 3. Caso contrário, caminhe sempre à esquerda enquanto o filho esquerdo não for NULL

Sucessor em uma ABB



Sucessor em uma ABB



Sucessor em uma ABB

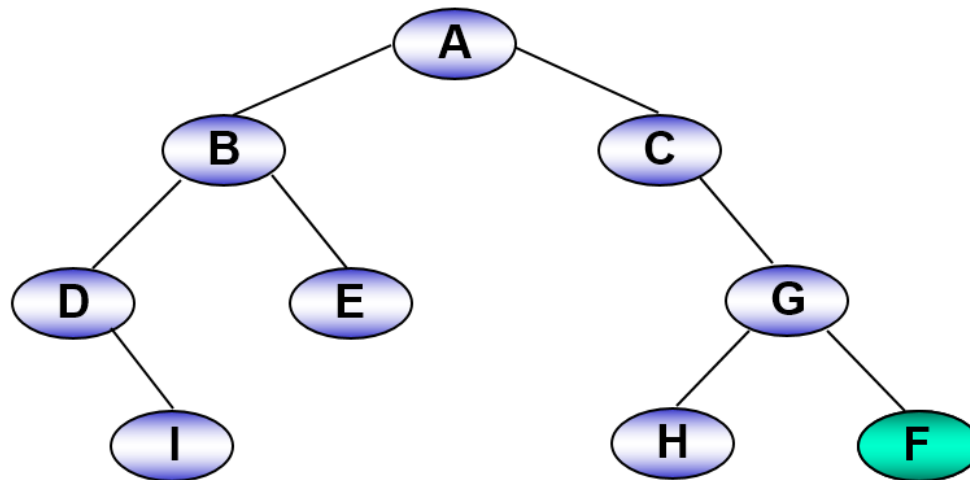
1. Se o nó for NULL return NULL
2. Se o nó tiver uma sub-árvore à direita, retorne o mínimo dela
3. Suba na árvore procurando o primeiro ancestral que seja maior que seu filho, ou seja, o nó corrente vai estar na sua sub-árvore à esquerda. Se nessa busca você chegar na raiz (ancestral NULL), retorne NULL.

Percursos em Árvores Binárias

- Um percurso consiste em se percorrer todos os nós de uma árvore em determinada sequência com o objetivo de consultar ou alterar suas informações
- Durante um percurso, ao se acessar um nó para consultar ou alterar suas informações, nós dizemos que estamos **visitando o nó**
- Os percursos mais famosos são:
 - Pré-ordem, pós-ordem, ordem simétrica, em largura

Percurso em pré-ordem (profundidade)

- Para percorrer uma árvore binária em pré-ordem:
 1. Visitamos a raiz da árvore
 2. Percorremos a subárvore esquerda
 3. Percorremos a subárvore direita



Seqüência: A B D I E C G H F

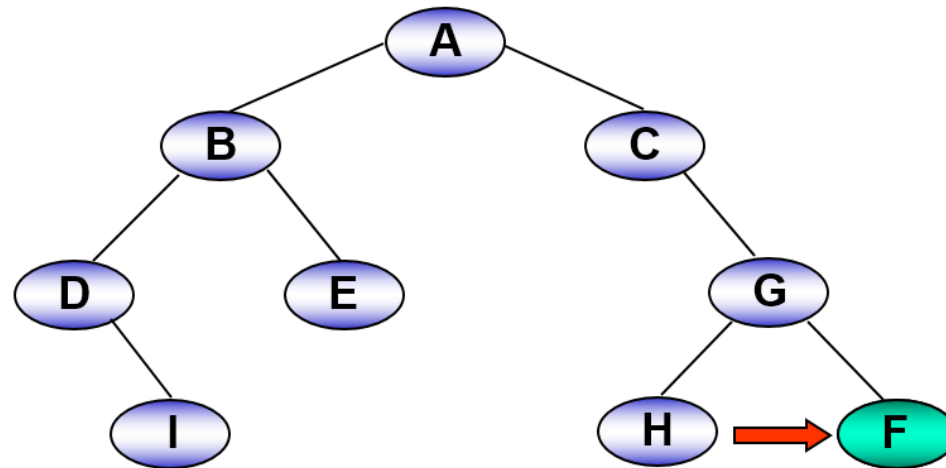
Percurso em pré-ordem

- Implementação

```
void preOrdem(NoArvoreBin no) {  
    if (no != null) {  
        // Visite o no  
        preOrdem(no.esq) ;  
        preOrdem(no.dir) ;  
    }  
}
```


Percurso em ordem simétrica (em ordem)

- Para percorrer uma árvore binária em ordem simétrica:
 1. Percorremos a subárvore esquerda
 2. Visitamos a raiz da árvore
 3. Percorremos a subárvore direita



Seqüência: D I B E A C H G F

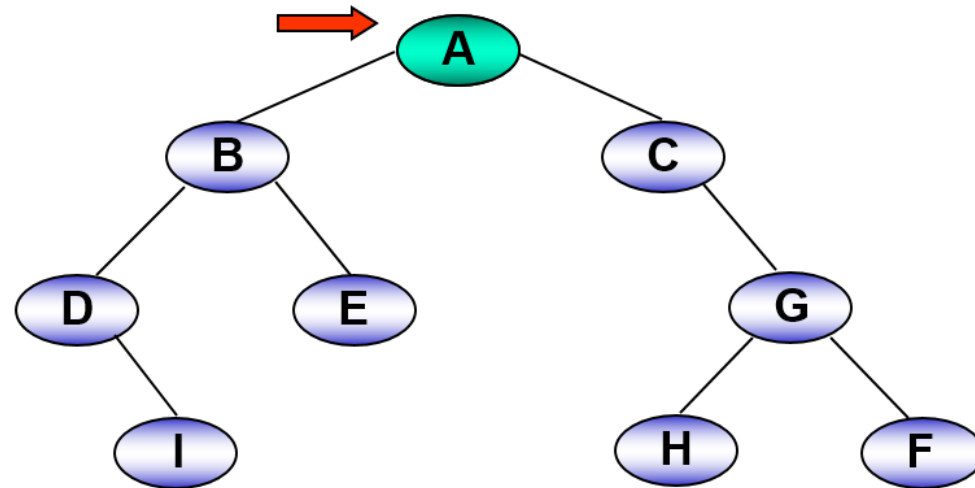
Percurso em ordem simétrica

- Implementação

```
void simetrica(NoArvoreBin no) {  
    if (no != null) {  
        simetrica(no.esq);  
        // Visite o no  
        simetrica(no.dir);  
    }  
}
```

Percurso em ordem pós-ordem

- Para percorrer uma árvore binária em pós-ordem:
 1. Percorremos a subárvore esquerda
 2. Percorremos a subárvore direita
 3. Visitamos a raiz da árvore



Seqüência: I D E B H F G C A

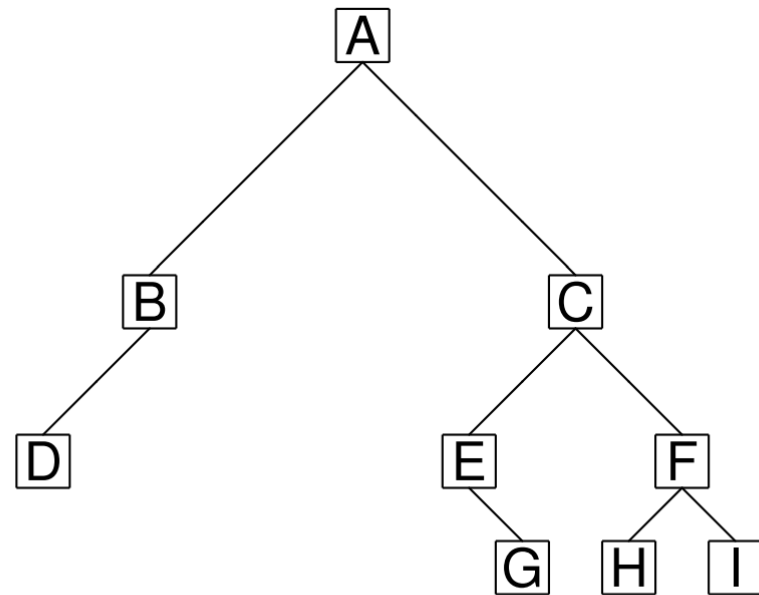
Percurso em ordem pós-ordem

- Implementação

```
void posOrdem(NoArvoreBin no) {  
    if (no != null) {  
        posOrdem(no.esq) ;  
        posOrdem(no.dir) ;  
        // Visite o no  
    }  
}
```

Outros percursos

- Há ainda outras formas usuais para percorrer uma árvore
- O percurso **em largura** (ou **em nível**) percorre a árvore em ordem crescente de seus níveis e, em cada nível, da esquerda para a direita



Percurso em largura: A B C D E F G H I

Dúvidas?

Exercício

Exercício ABB

1. Adicione os 3 percursos (pré, pós e em ordem) na árvore binária implementada na aula anterior
2. Altere o método `Collection<T> obterTodos()`
 1. `Collection<T> obterTodosPreOrdem();`
 2. `Collection<T> obterTodosPosOrdem();`
 3. `Collection<T> obterTodosEmOrdem();`