

Árvores

Programação II – Engenharia de Telecomunicações

Prof. Emerson Ribeiro de Mello

mello@ifsc.edu.br

Licenciamento



Slides licenciados sob [Creative Commons "Atribuição 4.0 Internacional"](https://creativecommons.org/licenses/by/4.0/)

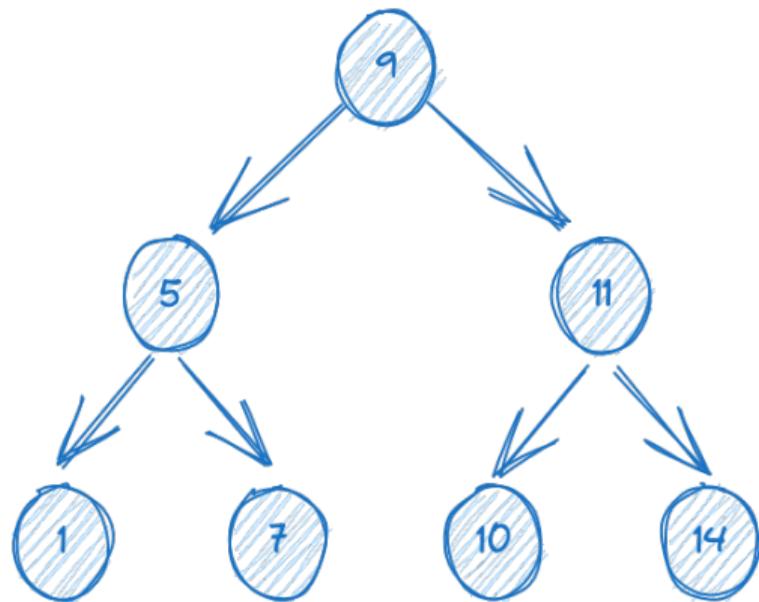
Sumário

- 1 Definições gerais
- 2 Árvore de busca binária
- 3 Exercícios

Definições gerais

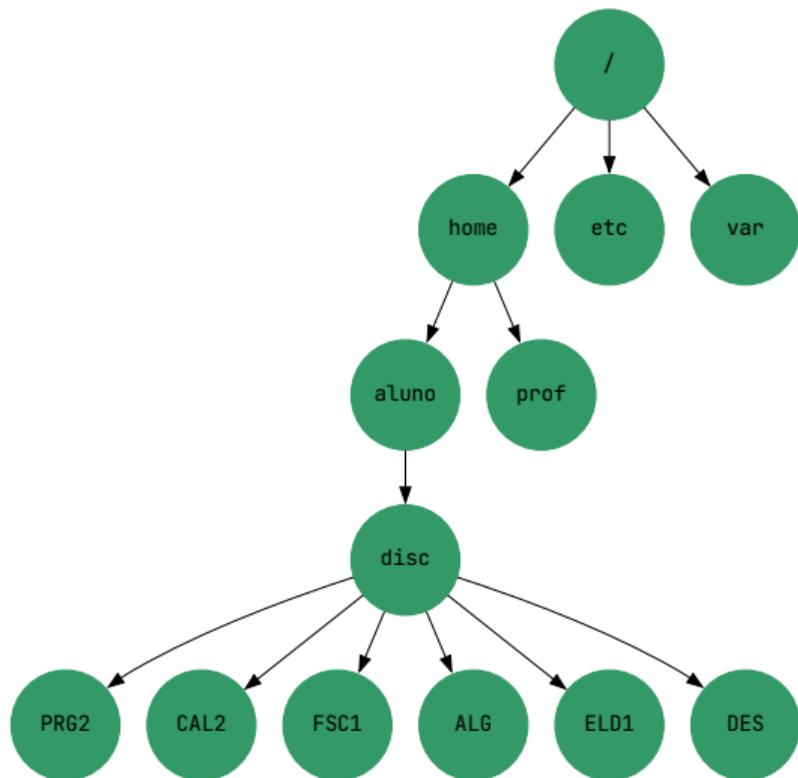
Árvores

- Estrutura de dados não linear, hierárquica e recursiva
- Formada por um conjunto de **nós**, sendo que um deles é a **raiz**
- Exemplo de aplicações:
 - sistemas de arquivos
 - banco de dados
 - *blockchain*



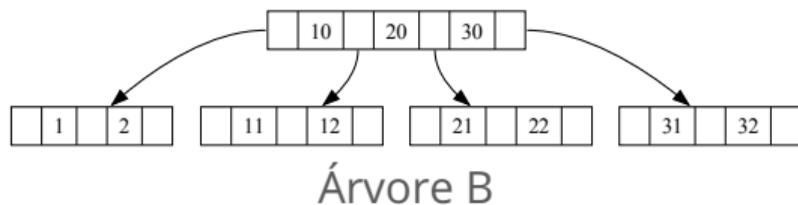
Árvores

- Estrutura de dados não linear, hierárquica e recursiva
- Formada por um conjunto de **nós**, sendo que um deles é a **raiz**
- Exemplo de aplicações:
 - sistemas de arquivos
 - banco de dados
 - *blockchain*



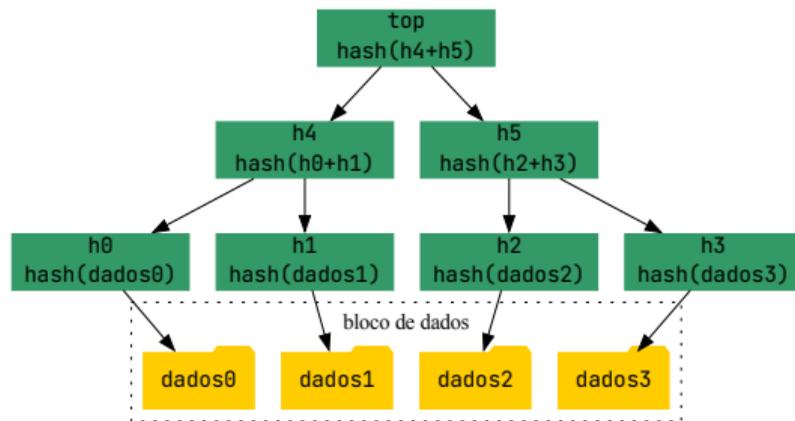
Árvores

- Estrutura de dados não linear, hierárquica e recursiva
- Formada por um conjunto de **nós**, sendo que um deles é a **raiz**
- Exemplo de aplicações:
 - sistemas de arquivos
 - **banco de dados**
 - *blockchain*



Árvores

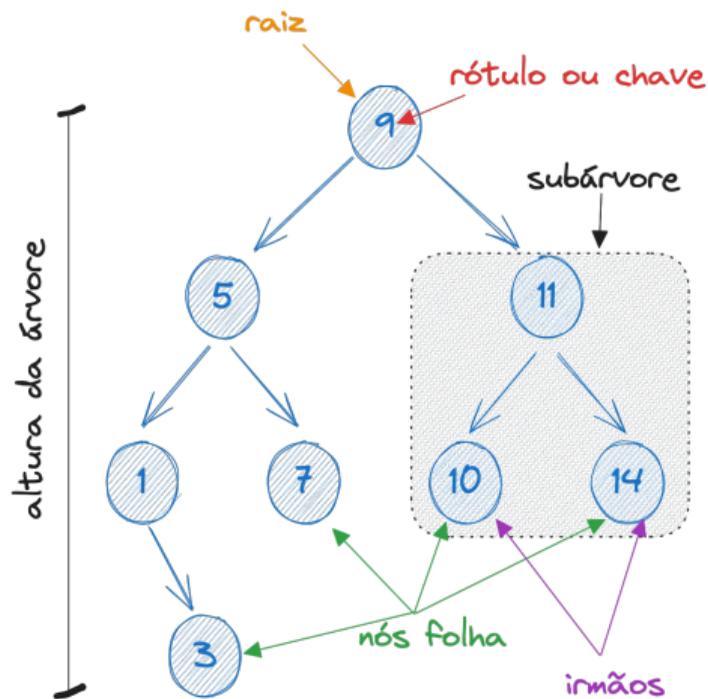
- Estrutura de dados não linear, hierárquica e recursiva
- Formada por um conjunto de **nós**, sendo que um deles é a **raiz**
- Exemplo de aplicações:
 - sistemas de arquivos
 - banco de dados
 - *blockchain*



Árvore de Merkle

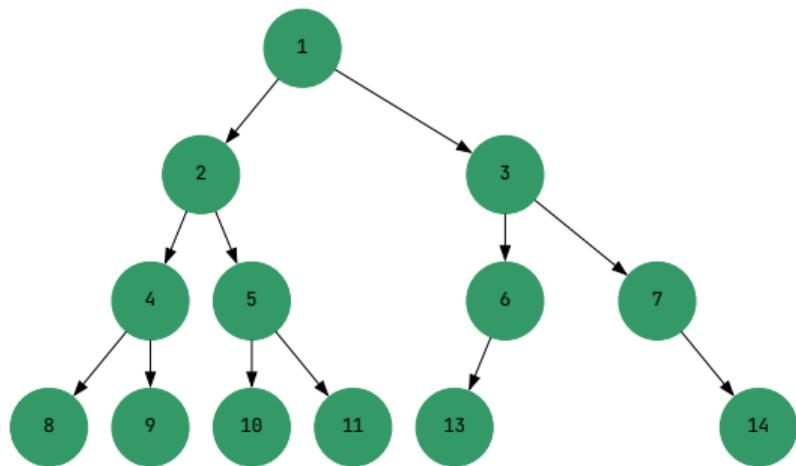
Árvores

Definições gerais



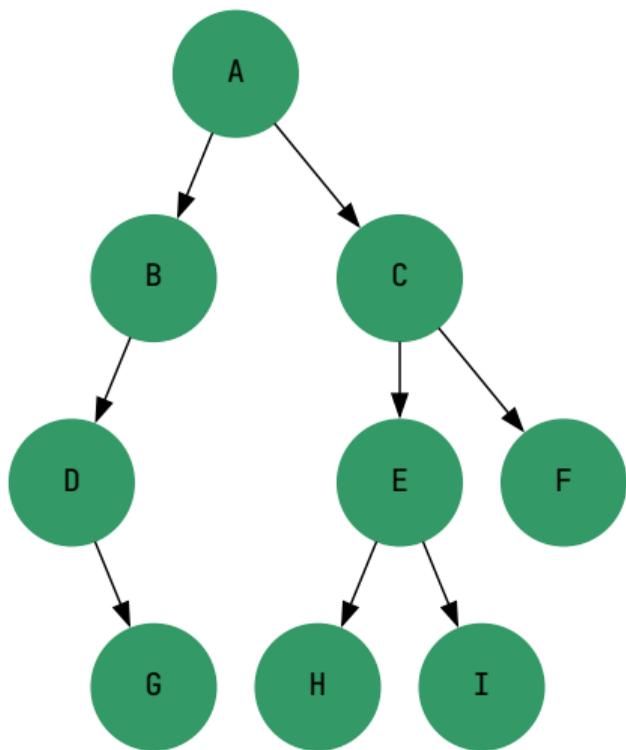
- **Nó raiz:** nó que não possui nó pai
- **Nó irmão:** nós que possuem o mesmo nó pai
- **Nó folha:** nó que não possui filhos
- **Grau de um nó:** número de filhos de um nó
- **Grau de uma árvore:** máximo grau de seus nós
- **Altura:** número de níveis (nível inicia em 0)
- **Tamanho:** número de nós

Árvore binária



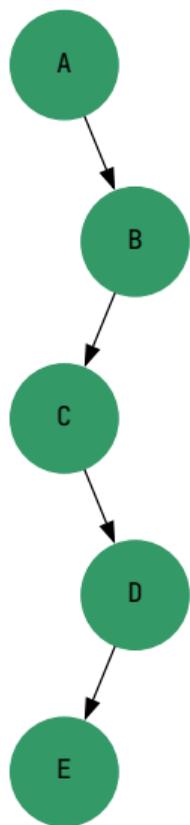
- Consiste de uma árvore de grau 2, ou seja, todo nó tem **no máximo dois filhos**
 - **filho esquerdo e filho direito**
- Um filho esquerdo pode existir sem um filho direito e vice-versa
- É o tipo de árvore com maior aplicação em computação

Árvore binária



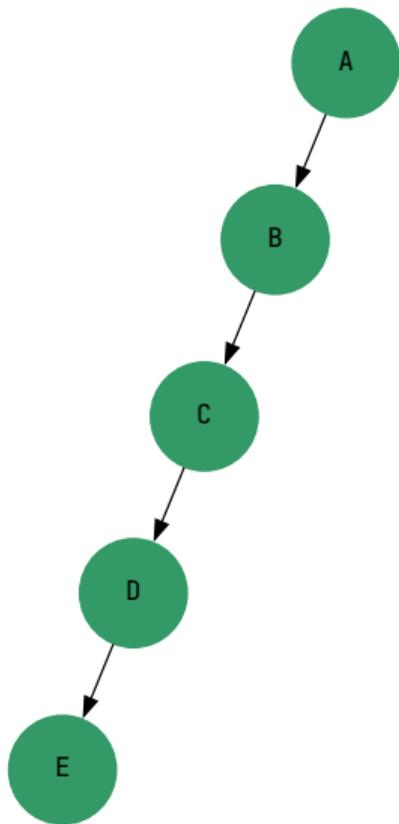
- Consiste de uma árvore de grau 2, ou seja, todo nó tem **no máximo dois filhos**
 - **filho esquerdo e filho direito**
- Um filho esquerdo pode existir sem um filho direito e vice-versa
- É o tipo de árvore com maior aplicação em computação

Árvore binária



- Consiste de uma árvore de grau 2, ou seja, todo nó tem **no máximo dois filhos**
 - **filho esquerdo** e **filho direito**
- Um filho esquerdo pode existir sem um filho direito e vice-versa
- É o tipo de árvore com maior aplicação em computação

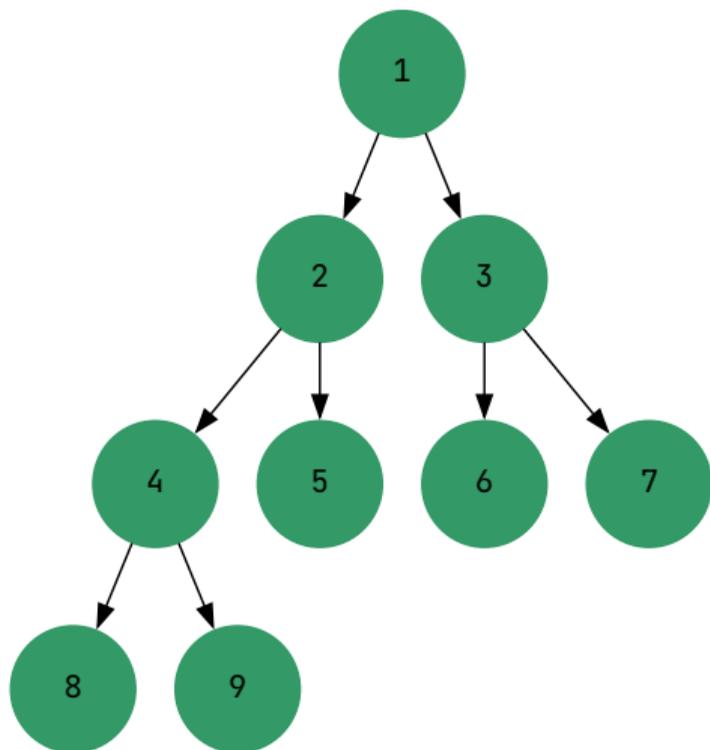
Árvore binária



- Consiste de uma árvore de grau 2, ou seja, todo nó tem **no máximo dois filhos**
 - **filho esquerdo** e **filho direito**
- Um filho esquerdo pode existir sem um filho direito e vice-versa
- É o tipo de árvore com maior aplicação em computação

Classificações de Árvores binárias

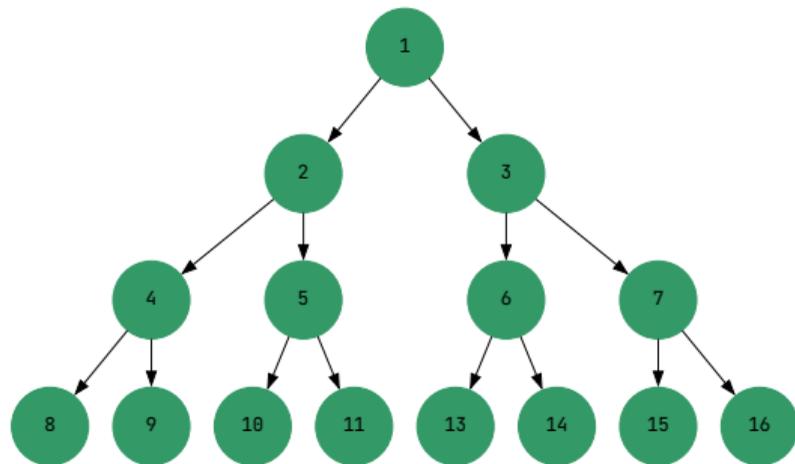
Árvore binária completa



- **Árvore binária completa:** árvore binária em que todos os níveis estão completos, exceto possivelmente o último
- **Árvore binária cheia:** árvore binária em que todos os níveis estão completos
- **Árvore estritamente binária:** árvore binária em que cada nó possui dois filhos ou nenhum

Classificações de Árvores binárias

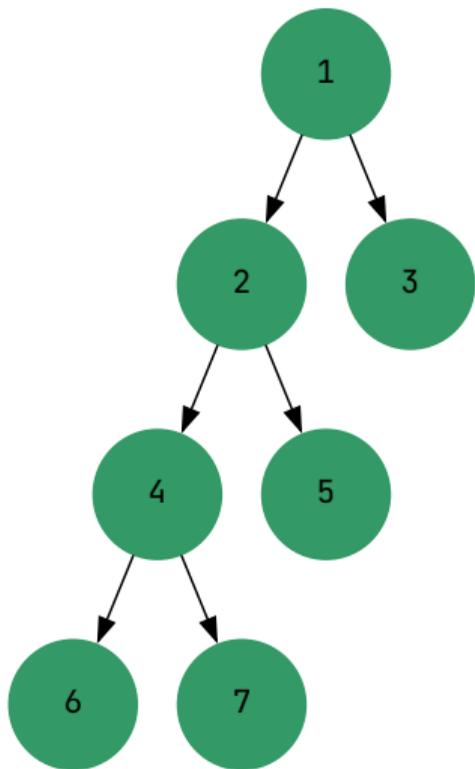
Árvore binária cheia



- **Árvore binária completa:** árvore binária em que todos os níveis estão completos, exceto possivelmente o último
- **Árvore binária cheia:** árvore binária em que todos os níveis estão completos
- **Árvore estritamente binária:** árvore binária em que cada nó possui dois filhos ou nenhum

Classificações de Árvores binárias

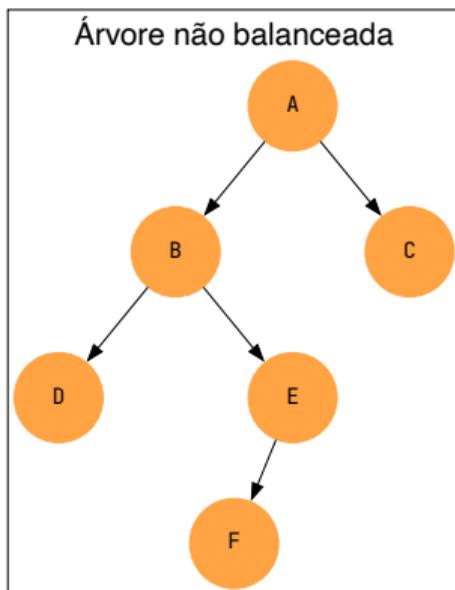
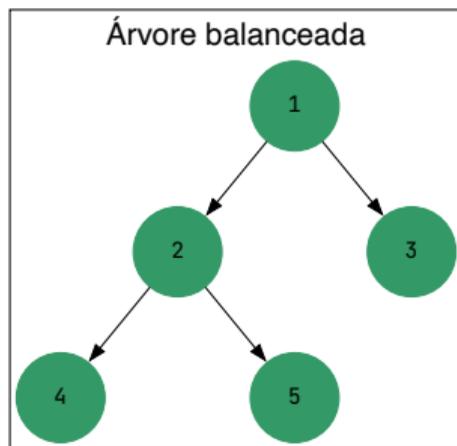
Árvore binária estritamente binária



- **Árvore binária completa:** árvore binária em que todos os níveis estão completos, exceto possivelmente o último
- **Árvore binária cheia:** árvore binária em que todos os níveis estão completos
- **Árvore estritamente binária:** árvore binária em que cada nó possui dois filhos ou nenhum

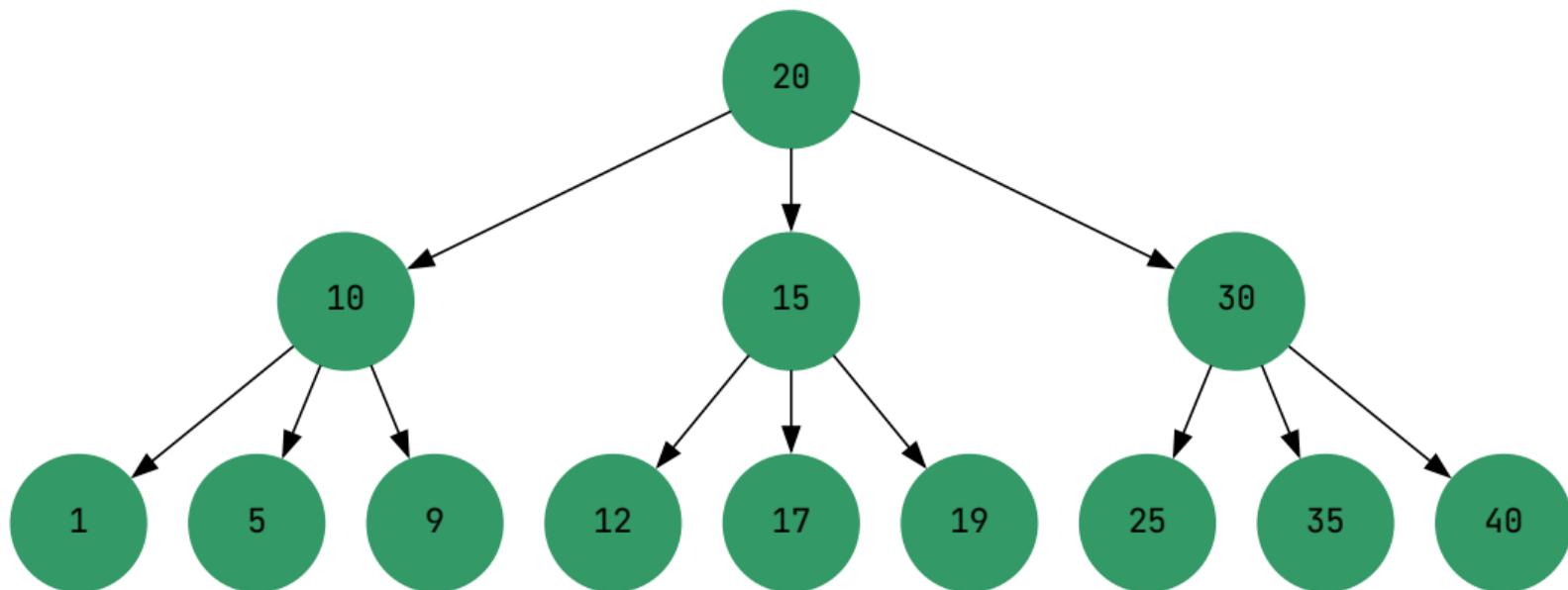
Árvore binária balanceada

A diferença entre as alturas das subárvores esquerda e direita de qualquer nó é no máximo 1



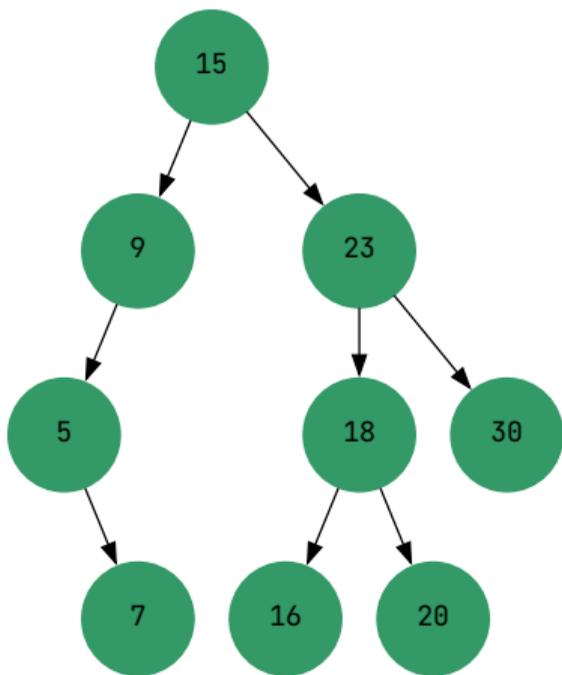
Árvore m-ária

- Uma árvore m-ária é uma árvore em que cada nó possui no máximo m filhos, sendo $m \geq 2$
- Uma árvore ternária é uma árvore m-ária em que $m = 3$



Árvore de busca binária

Árvore de busca binária ou Árvore de pesquisa binária



- Para cada nó, todos os nós da **subárvore**
 - **esquerda** possuem valores menores que o valor do nó
 - **direita** possuem valores maiores que o valor do nó
- Permite armazenar e pesquisar dados de forma eficiente
 - Caso médio: $O(\log n)$
 - Pior caso: $O(n)$
 - Complexidade de espaço: $O(n)$

Árvore de busca binária

Implementação

```
typedef struct no {
    int valor;
    struct no *esquerda;
    struct no *direita;
} no_t;

no_t *criar_no(int valor){
    no_t *no = (no_t *) malloc(sizeof(no_t));
    no->valor = valor;
    no->esquerda = no->direita = NULL;
    return no;
}

void destruir_no(no_t *no) {
    if (no != NULL) {
        destruir_no(no->esquerda);
        destruir_no(no->direita);
        free(no);
    }
}
```

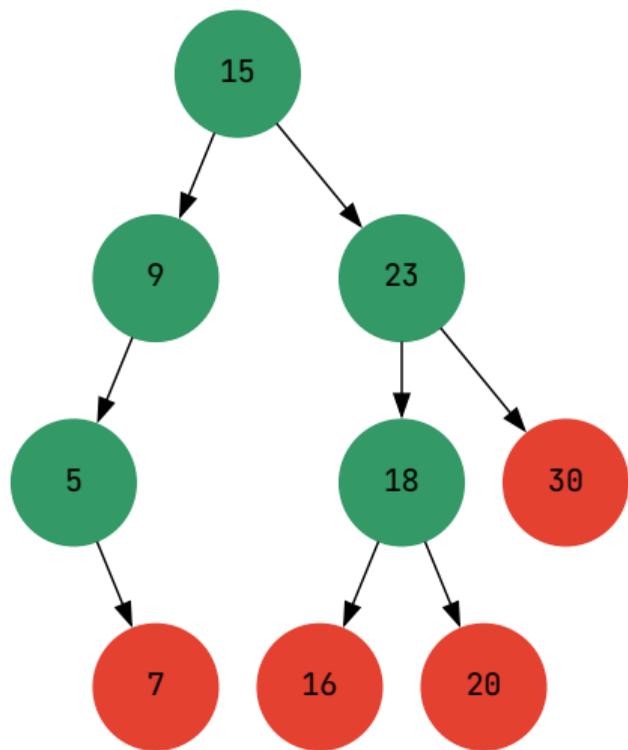
Árvore de busca binária

Implementação das funções para inserir e buscar

```
// todo nó é raiz de uma subárvore
no_t *inserir_valor(no_t *raiz, int valor){
    if (raiz == NULL) {
        return criar_no(valor);
    } else if (valor < raiz->valor) {
        raiz->esquerda = inserir_valor(raiz->esquerda, valor);
    } else if (valor > raiz->valor) {
        raiz->direita = inserir_valor(raiz->direita, valor);
    }
    return raiz;
}

bool busca(no_t *raiz, int valor){
    if (raiz == NULL) return false;
    if (valor == raiz->valor) return true;
    if (valor < raiz->valor) return busca(raiz->esquerda, valor);
    return busca(raiz->direita, valor);
}
```

Como excluir um nó de uma Árvore de busca binária?

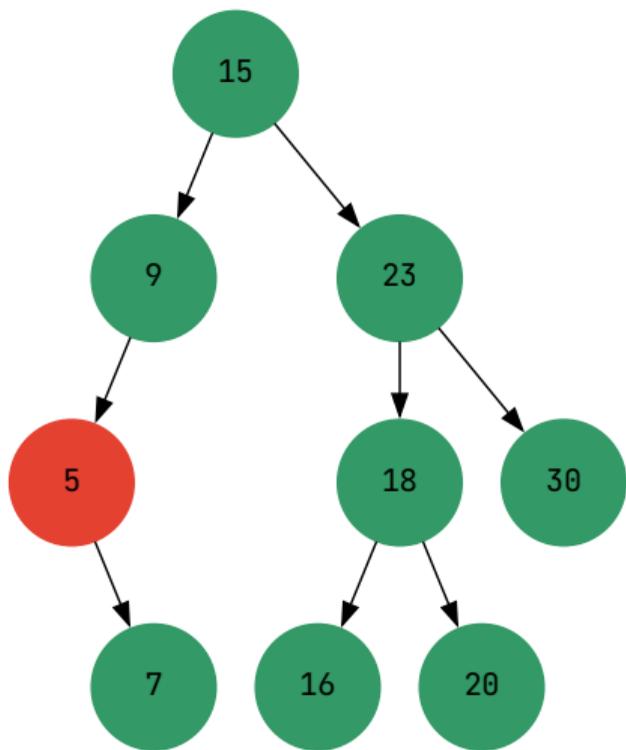


```
no_t *remove_valor(no_t *raiz, int valor){  
    if (raiz == NULL) return raiz;  
    if (valor < raiz->valor) {  
        // raiz->esquerda = remover...  
    } else if (valor > raiz->valor) {  
        // raiz->direita = remover...  
    } else {  
        // IF nó folha ou nó com um filho  
        // ELSE nó com dois filhos  
    }  
    return raiz;  
}
```

■ Nó folha

- basta remover o nó

Como excluir um nó de uma Árvore de busca binária?

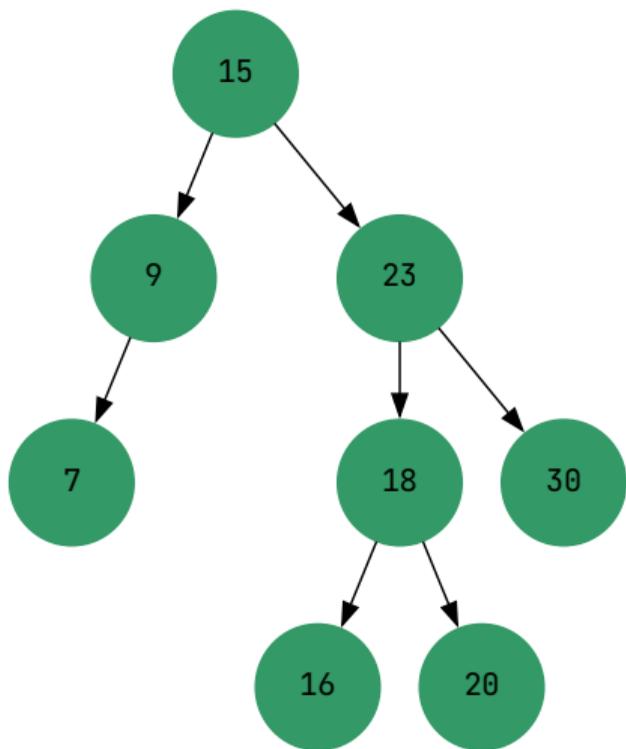


```
no_t *remove_valor(no_t *raiz, int valor){
    if (raiz == NULL) return raiz;
    if (valor < raiz->valor) {
        // raiz->esquerda = remover...
    } else if (valor > raiz->valor) {
        // raiz->direita = remover...
    } else {
        // IF nó folha ou nó com um filho
        // ELSE nó com dois filhos
    }
    return raiz;
}
```

■ Nó com apenas um filho

- basta remover o nó e ligar o filho do nó a ser excluído ao pai do nó a ser excluído

Como excluir um nó de uma Árvore de busca binária?



```
no_t *remove_valor(no_t *raiz, int valor){  
    if (raiz == NULL) return raiz;  
    if (valor < raiz->valor) {  
        // raiz->esquerda = remover...  
    } else if (valor > raiz->valor) {  
        // raiz->direita = remover...  
    } else {  
        // IF nó folha ou nó com um filho  
        // ELSE nó com dois filhos  
    }  
    return raiz;  
}
```

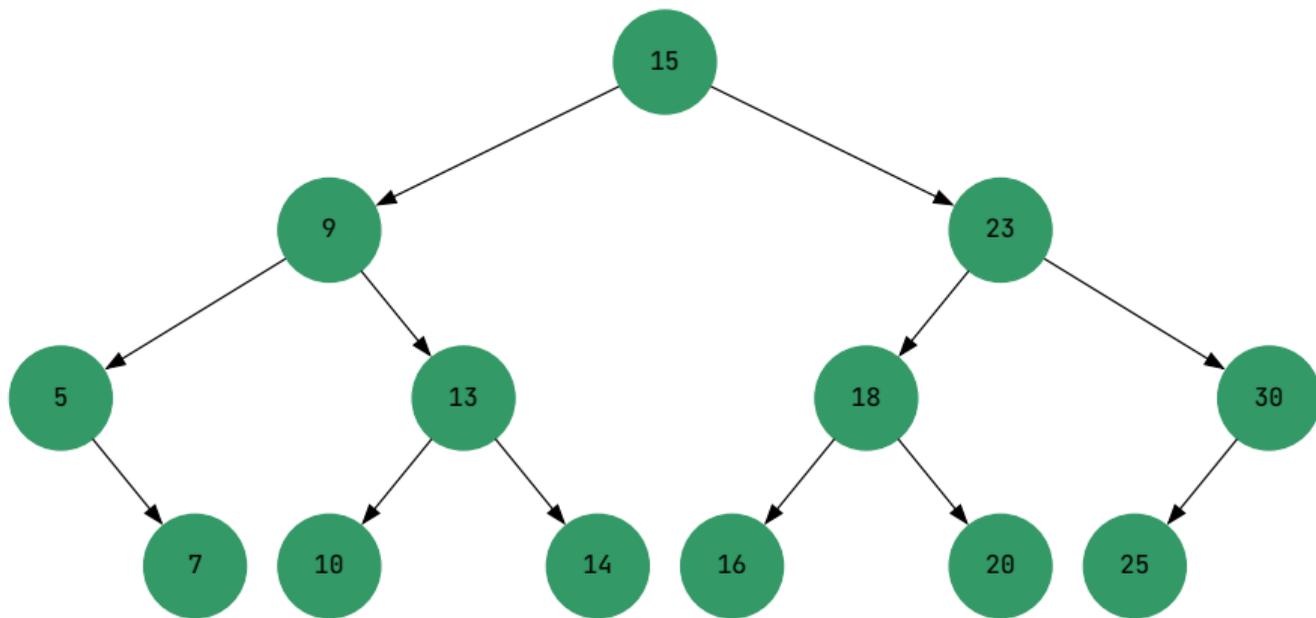
■ Nó com apenas um filho

- basta remover o nó e ligar o filho do nó a ser excluído ao pai do nó a ser excluído

Como excluir um nó de uma Árvore de busca binária?

Nó com dois filhos

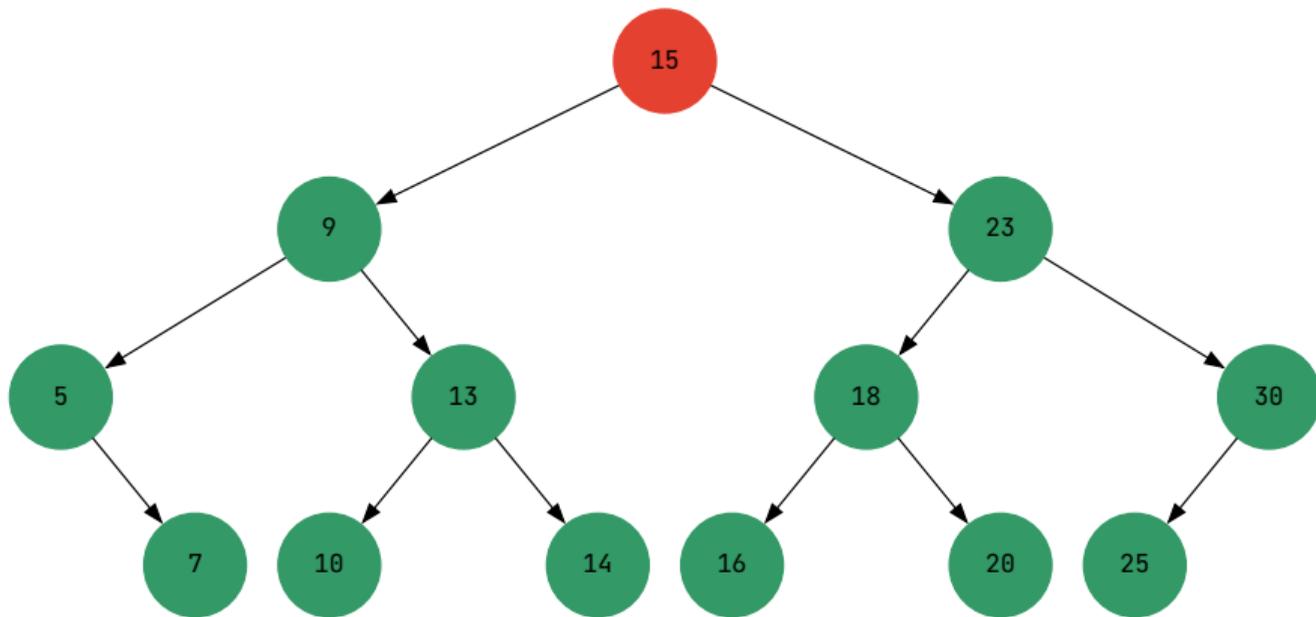
- O valor do nó a ser excluído é substituído pelo valor do nó mais à direita da subárvore esquerda (ou pelo valor do nó mais à esquerda da subárvore direita)



Como excluir um nó de uma Árvore de busca binária?

Nó com dois filhos

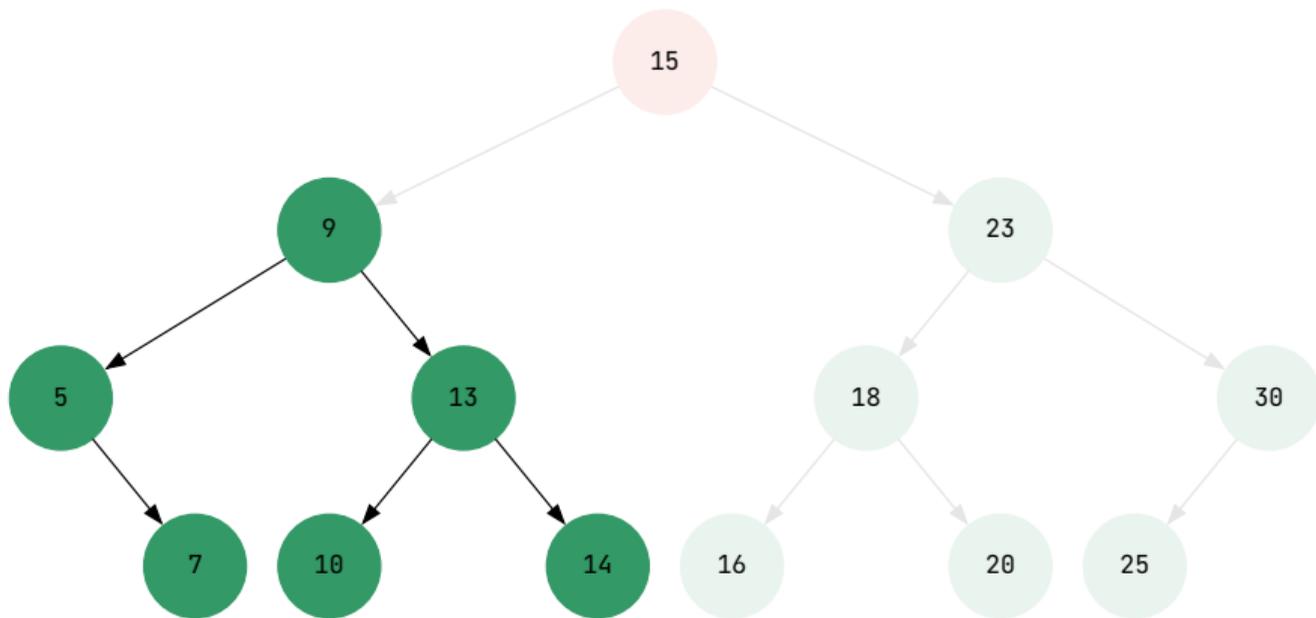
- O valor do nó a ser excluído é substituído pelo valor do nó mais à direita da subárvore esquerda (ou pelo valor do nó mais à esquerda da subárvore direita)



Como excluir um nó de uma Árvore de busca binária?

Nó com dois filhos

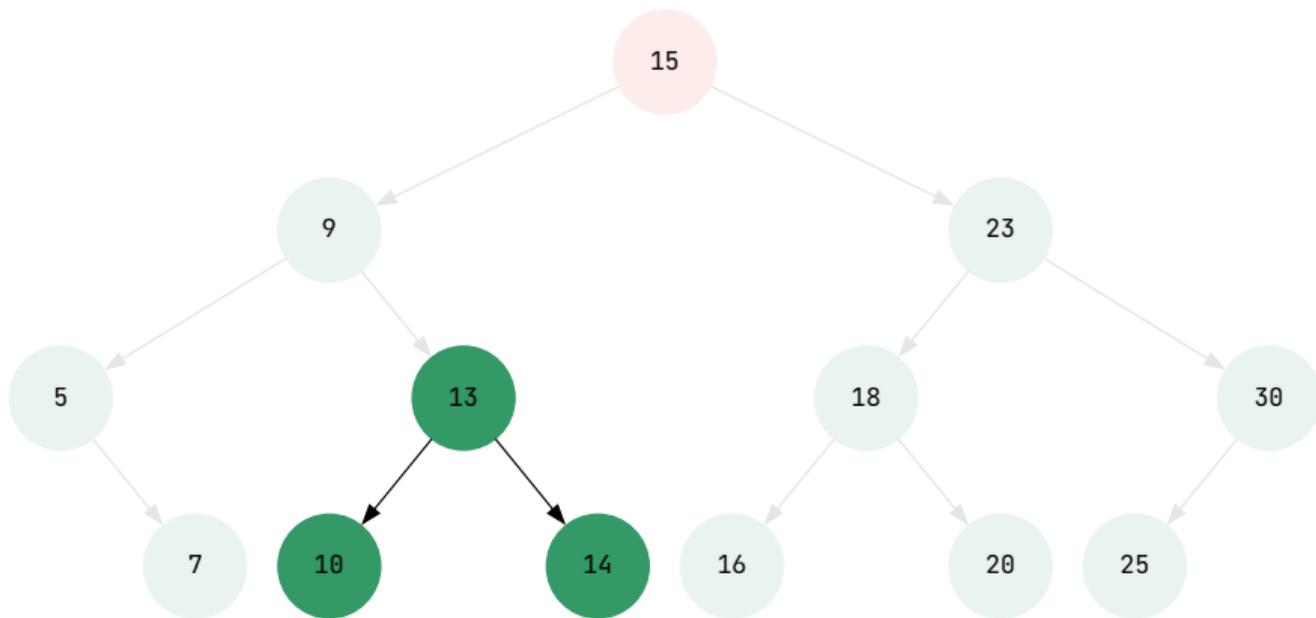
- O valor do nó a ser excluído é substituído pelo valor do nó mais à direita da subárvore esquerda (ou pelo valor do nó mais à esquerda da subárvore direita)



Como excluir um nó de uma Árvore de busca binária?

Nó com dois filhos

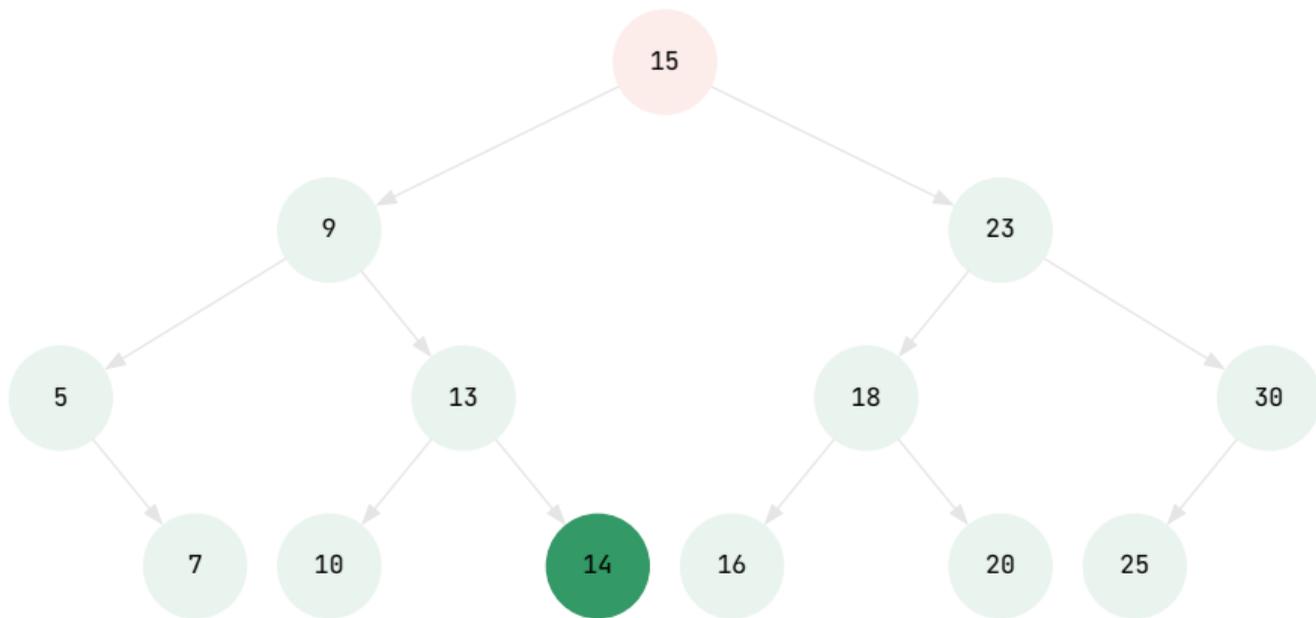
- O valor do nó a ser excluído é substituído pelo valor do nó mais à direita da subárvore esquerda (ou pelo valor do nó mais à esquerda da subárvore direita)



Como excluir um nó de uma Árvore de busca binária?

Nó com dois filhos

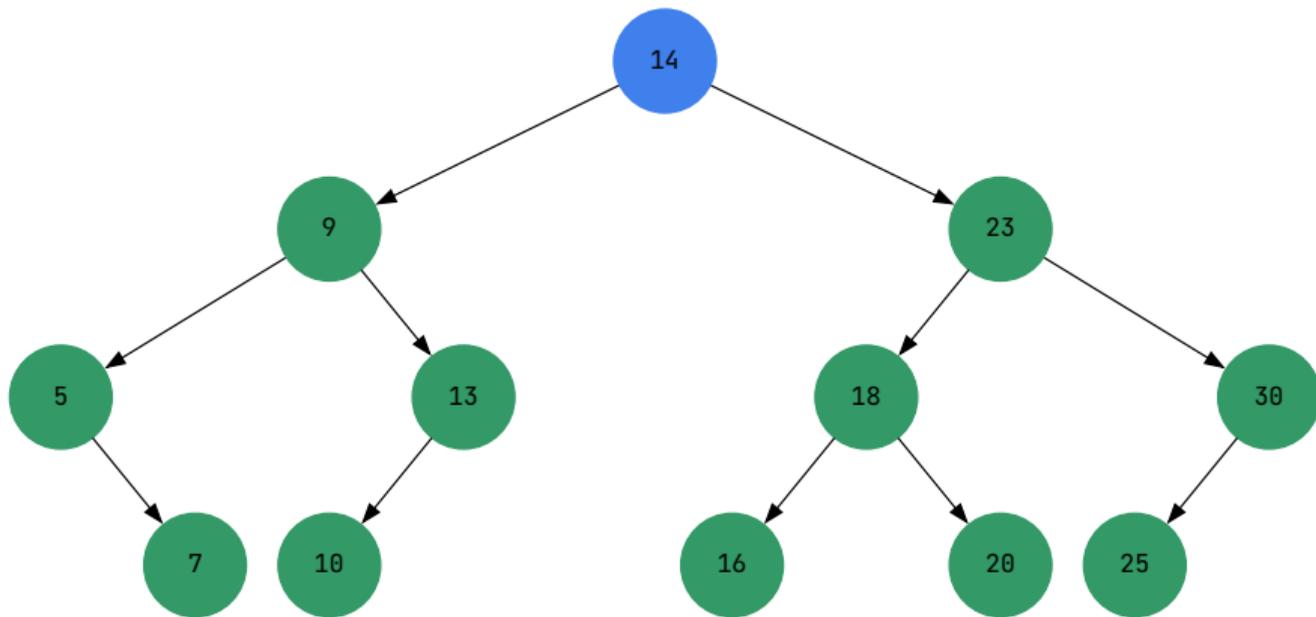
- O valor do nó a ser excluído é substituído pelo valor do nó mais à direita da subárvore esquerda (ou pelo valor do nó mais à esquerda da subárvore direita)



Como excluir um nó de uma Árvore de busca binária?

Nó com dois filhos

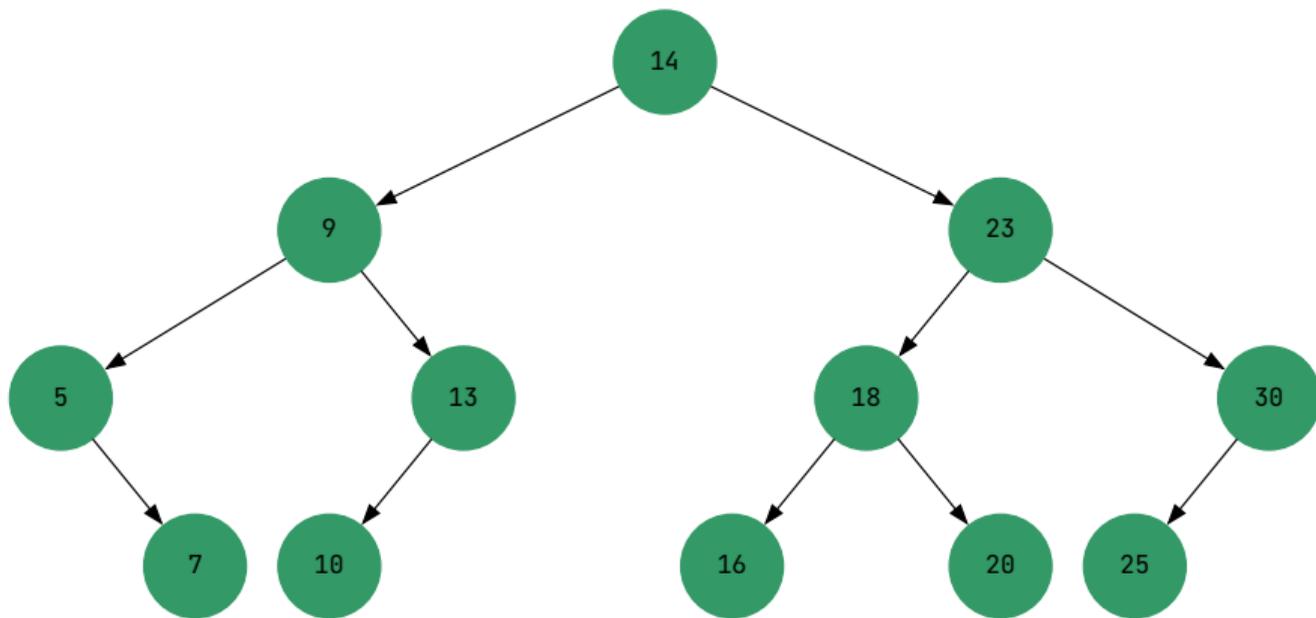
- O valor do nó a ser excluído é substituído pelo valor do nó mais à direita da subárvore esquerda (ou pelo valor do nó mais à esquerda da subárvore direita)



Como excluir um nó de uma Árvore de busca binária?

Nó com dois filhos

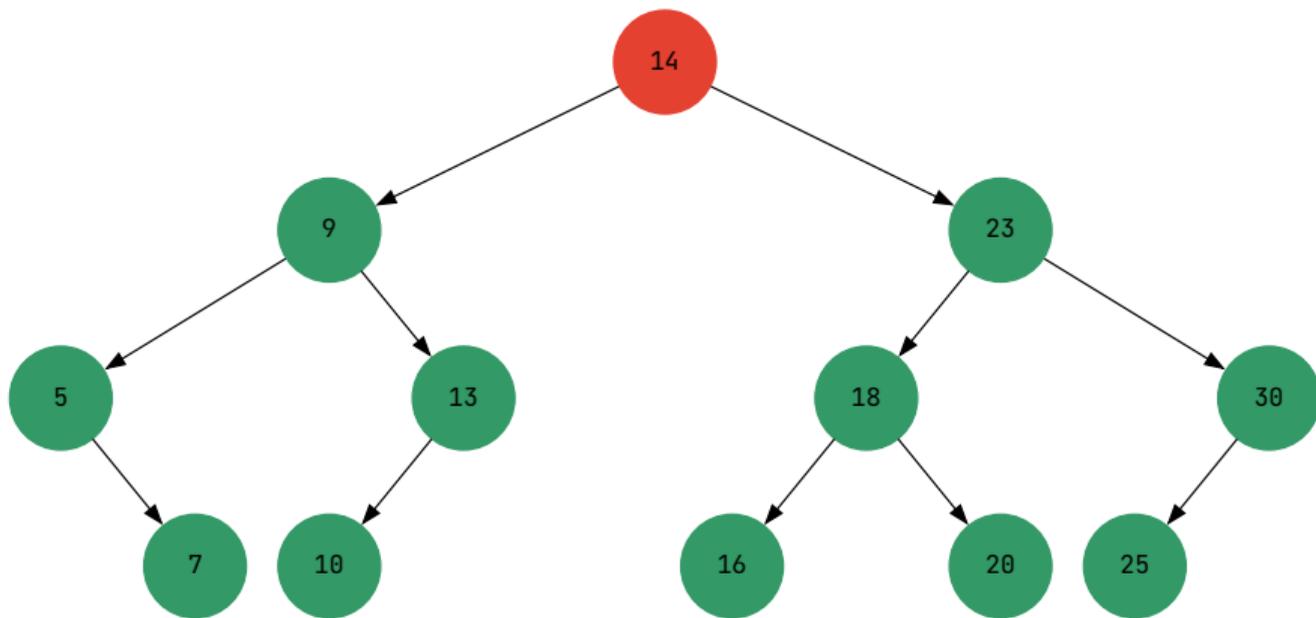
- O valor do nó a ser excluído é substituído pelo valor do nó mais à direita da subárvore esquerda (ou pelo valor do nó mais à esquerda da subárvore direita)



Como excluir um nó de uma Árvore de busca binária?

Nó com dois filhos

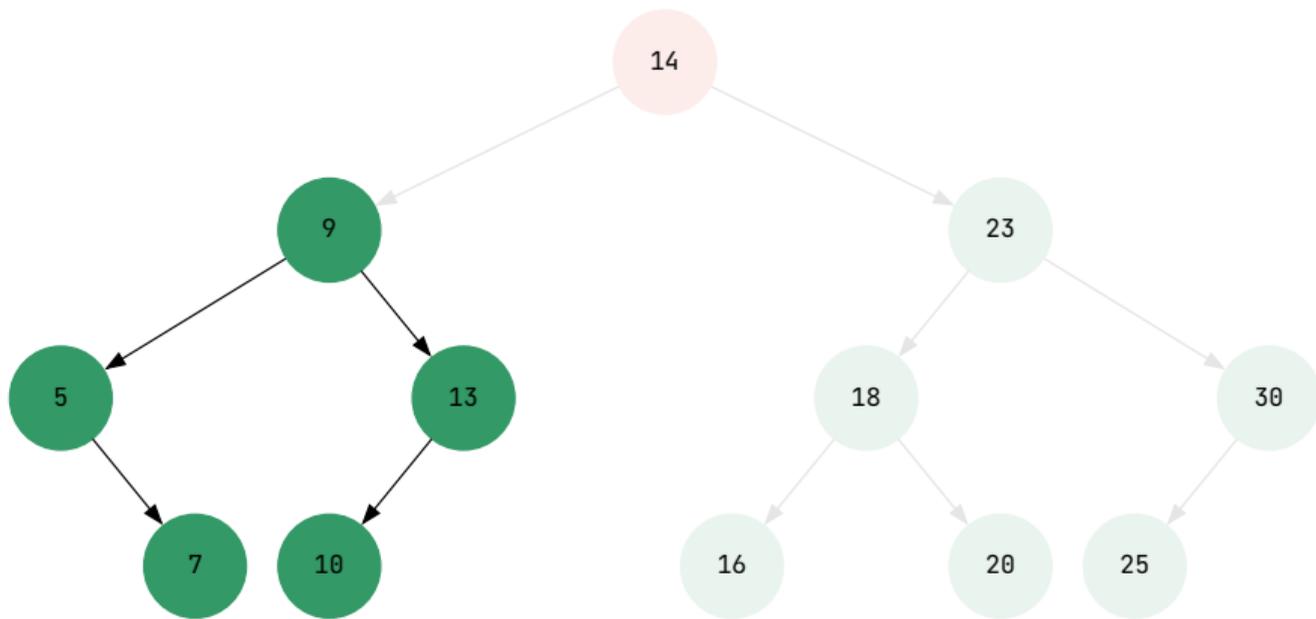
- O valor do nó a ser excluído é substituído pelo valor do nó mais à direita da subárvore esquerda (ou pelo valor do nó mais à esquerda da subárvore direita)



Como excluir um nó de uma Árvore de busca binária?

Nó com dois filhos

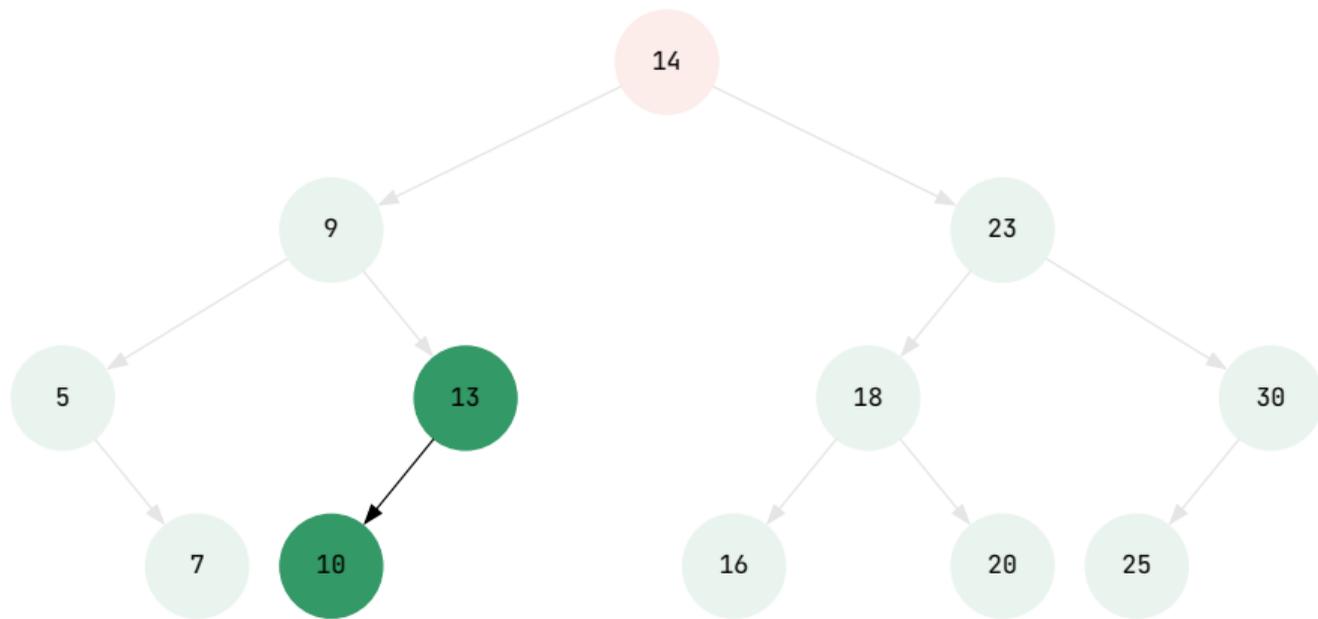
- O valor do nó a ser excluído é substituído pelo valor do nó mais à direita da subárvore esquerda (ou pelo valor do nó mais à esquerda da subárvore direita)



Como excluir um nó de uma Árvore de busca binária?

Nó com dois filhos

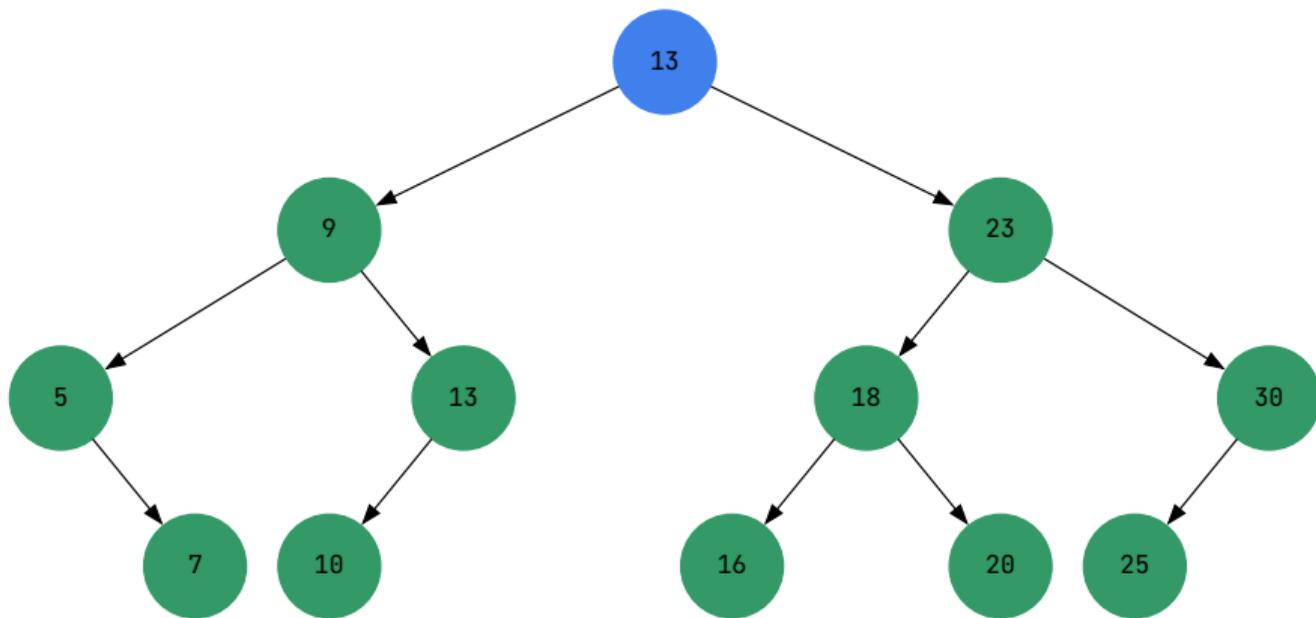
- O valor do nó a ser excluído é substituído pelo valor do nó mais à direita da subárvore esquerda (ou pelo valor do nó mais à esquerda da subárvore direita)



Como excluir um nó de uma Árvore de busca binária?

Nó com dois filhos

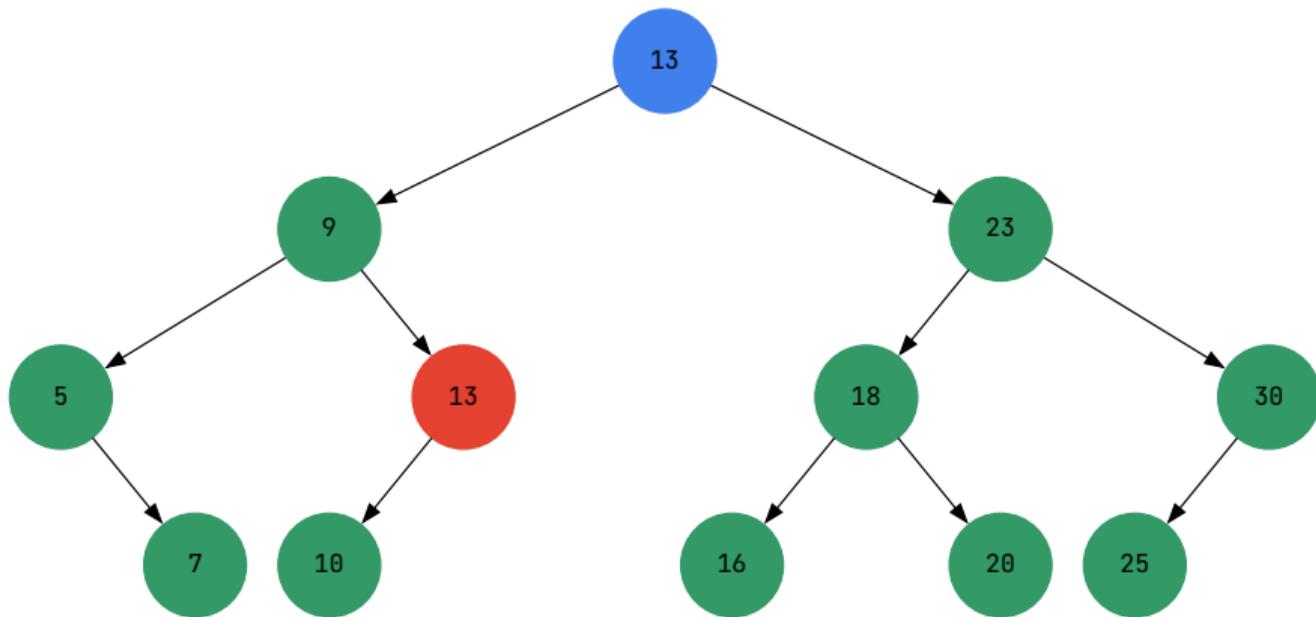
- O valor do nó a ser excluído é substituído pelo valor do nó mais à direita da subárvore esquerda (ou pelo valor do nó mais à esquerda da subárvore direita)



Como excluir um nó de uma Árvore de busca binária?

Nó com dois filhos

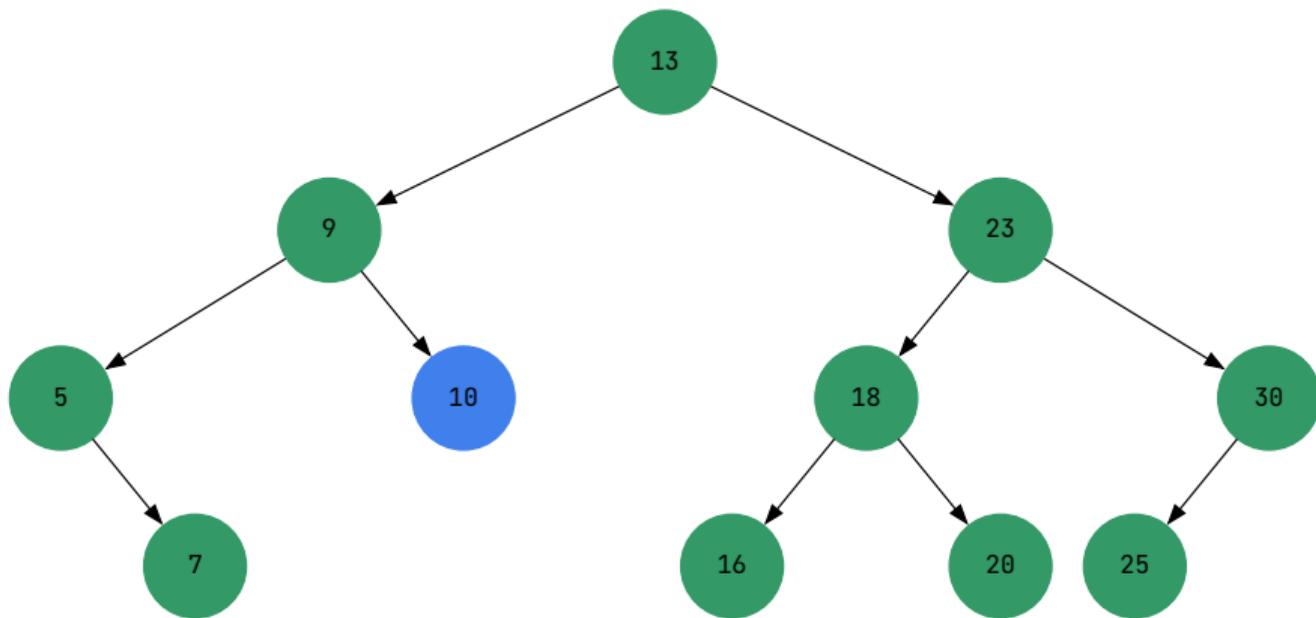
- O valor do nó a ser excluído é substituído pelo valor do nó mais à direita da subárvore esquerda (ou pelo valor do nó mais à esquerda da subárvore direita)



Como excluir um nó de uma Árvore de busca binária?

Nó com dois filhos

- O valor do nó a ser excluído é substituído pelo valor do nó mais à direita da subárvore esquerda (ou pelo valor do nó mais à esquerda da subárvore direita)



Exercícios

Exercício 1

Árvore de busca binária

- 1 Aproveite o código fornecido nesse conjunto de slides e implemente uma aplicação que gere n números inteiros de forma aleatória e insira-os em uma árvore de busca binária
 - Faça a tomada de tempo para criar a árvore e imprima o resultado
- 2 Faça uma função para encontrar o maior valor armazenado na árvore e imprima o resultado
 - Faça a tomada de tempo para encontrar o maior valor e imprima o resultado
- 3 Faça uma função para encontrar o menor valor armazenado na árvore e imprima o resultado
 - Faça a tomada de tempo para encontrar o menor valor e imprima o resultado

Exercício 2

Árvore de busca binária

Evolua o código do exercício 1

- 1 Permita ao usuário entrar com um número inteiro e verifique se esse número está na árvore
 - Caso esteja, imprima seus filhos e o nível em que ele se encontra
 - Faça a tomada de tempo para a busca e imprima o resultado
- 2 Permita ao usuário entrar com um número inteiro e remova-o da árvore

Referências

Aula baseada em

-  CORMEN, Thomas H. et al. **Algoritmos: teoria e prática**. LTC, 2012. Disponível em: <<https://app.minhabiblioteca.com.br/reader/books/9788595158092>>.
-  LAGO PEREIRA, Silvio do. **Estruturas de Dados em C - Uma Abordagem Didática**. Saraiva, 2016. ISBN 9788536517254. Disponível em: <<https://app.minhabiblioteca.com.br/#/books/9788536517254>>. Acesso em: 1 nov. 2023.
-  SZWARCFITER, Jayme Luiz; MARKENZON, Lilian. **Estruturas de dados e seus algoritmos**. LTC, 2010. Disponível em: <<https://app.minhabiblioteca.com.br/reader/books/978-85-216-2995-5>>.