Aula 5: Árvore de Busca Binária - BST

UFRR

DCC

DCC405-Estrutura de Dados II

Prof. Me. Acauan C. Ribeiro

Comparação entre estruturas

	Array (unsorted)	Linked List	Array (Sorted)	BST (balanced)
Search(x)	O(n)	O(n)	O(log n)	O(log n)
Insert(x)	O(1)	O(1)	O(n)	O(log n)
Remove(x)	O(n)	O(n)	O(n)	O(log n)

Comparação entre estruturas

	Array (unsorted)	Linked List	Array (Sorted)	BST (balanced)
Search(x)	O(n)	O(n)	O(log n)	O(log n)
Insert(x)	O(1)	O(1)	O(n)	O(log n)
Remove(x)	O(n)	O(n)	O(n)	O(log n)

→ Para **n registros**, se precisarmos realizar **n comparações** (pior caso)

→ Em um cenário que o **custo de 1 comparação** $= 10^{-6}$

Comparação entre estruturas

Search	
Insert	
Remove	

https://g1.globo.com/politica/noticia/2020/09/02/brasil-tem-125-milhoes-de-cpfs-ativos-a-mais-que-a-populacao-total-e-tcu-cobra-receita-federal.ghtml

→ Para **n registros**, se precisarmos realizar **n comparações** (pior caso)

→ Em um cenário que o **custo de 1 comparação** $= \mathbf{10^{-6}}$

Comparação entre estruturas

	Array (unsorted)	Linked List	Array (Sorted)	BST (balanced)
Search(x)	O(n)	O(n)	O(log n)	O(log n)
Insert(x)	O(1)	O(1)	O(n)	O(log n)
Remove(x)	O(n)	O(n)	O(n)	O(log n)

No relatório sobre as medidas tributárias adotadas pelo governo na pandemia de Covid-19, a Receita informou que há 223,8 milhões de CPFs ativos.

→ Para **n registros**, se precisarmos realizar **n comparações** (pior caso)

→ Em um cenário que o **custo de 1 comparação = 10^{-6} sec**

→ **n = $2{,}23\times10^8$ → $2{,}23\times10^8 \times 10^{-6}$ → $2{,}23 \times 10^2$ = 223s**

Comparação entre estruturas

	Array (unsorted)	Linked List	Array (Sorted)	BST (balanced)
Search(x)	O(n)	O(n)	O(log n)	O(log n)
Insert(x)	O(1)	O(1)	O(n)	O(log n)
Remove(x)	O(n)	O(n)	O(n)	O(log n)


globo.com g1 ge gshow globoplay

MENU g1 POLÍTICA

No relatório sobre as medidas tributárias adotadas pelo governo na **pandemia de Covid-19**, a Receita informou que há **223,8 milhões de CPFs ativos**.

→ Para **n registros**, se precisarmos realizar **log n comparações**

→ Em um cenário que o **custo de 1 comparação = 10^{-6} sec**

→ $n = 2{,}23 \times 10^{8} \rightarrow \log(2{,}23 \times 10^{8}) \times 10^{-6} \rightarrow 27{,}7325 \times 10^{-6} = 0{,}000027732$ s

Árvore de Busca Binária

Uma árvore de busca binária (BST) é uma árvore na qual todos os nós seguem as propriedades abaixo:

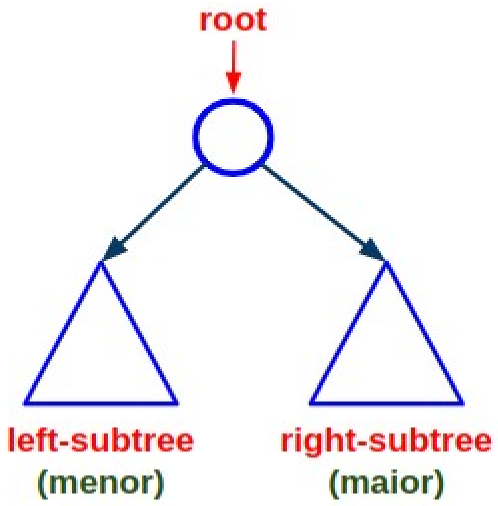
- O **valor da chave da subárvore esquerda** é **menor** que o valor da chave do nó pai (raiz).
- O **valor da chave da subárvore direita** é **maior** ou igual ao valor da chave de seu nó pai (raiz).

- **Binary Search Tree (BST)**, é um tipo específico de **árvore binária**, que implementa uma estrutura eficiente para organizar os dados, focando principalemnte na rapidez das ações de **busca** e **update**.

Árvore de Busca Binária

Seja x um nó em uma árvore de busca binária. Se y é um nó na subárvore esquerda de x, então $y.chave \leq x.chave$. Se y é um nó na subárvore direita de x, então $x.chave \geq y.chave$.

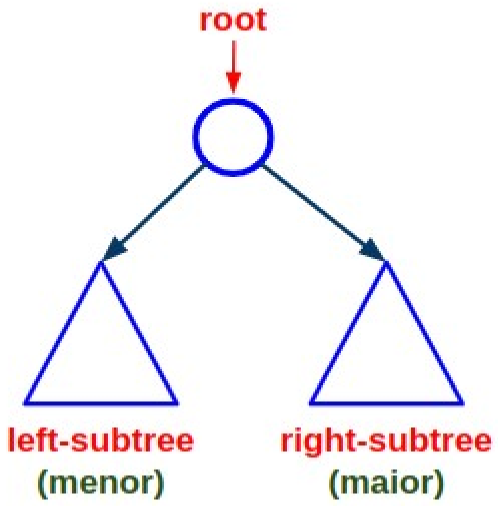
[Cormen et al., 2002]



Árvore de Busca Binária

Seja x um nó em uma árvore de busca binária. Se y é um nó na subárvore esquerda de x, então $y.chave \leq x.chave$. Se y é um nó na subárvore direita de x, então $x.chave \geq y.chave$.

[Cormen et al., 2002]



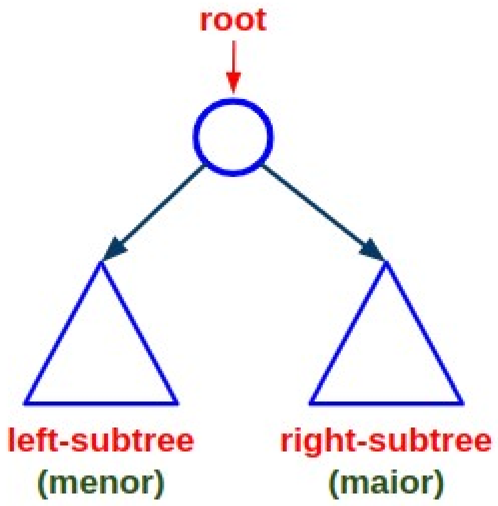
ou igual

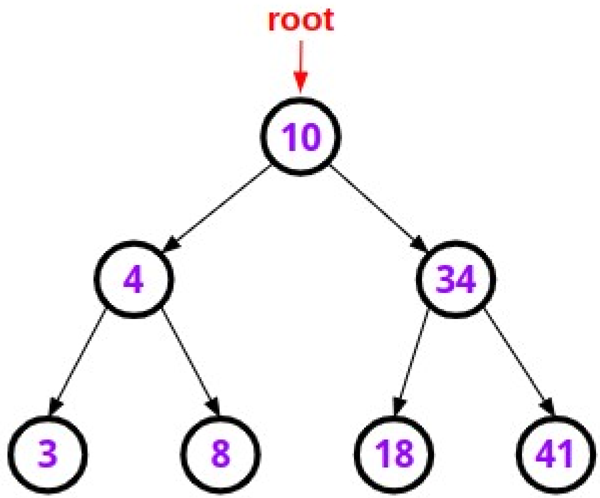
→ Neste caso a árvore aceita e trata chaves repetidas.

Árvore de Busca Binária

Seja x um nó em uma árvore de busca binária. Se y é um nó na subárvore esquerda de x, então $y.chave \leq x.chave$. Se y é um nó na subárvore direita de x, então $x.chave \geq y.chave$.

[Cormen et al., 2002]





Árvore de Busca Binária

Busca Binária

search(34)

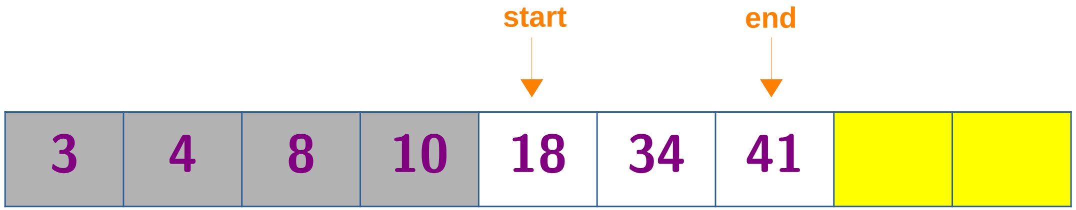
Árvore de Busca Binária

Busca Binária

search(34)

Árvore de Busca Binária

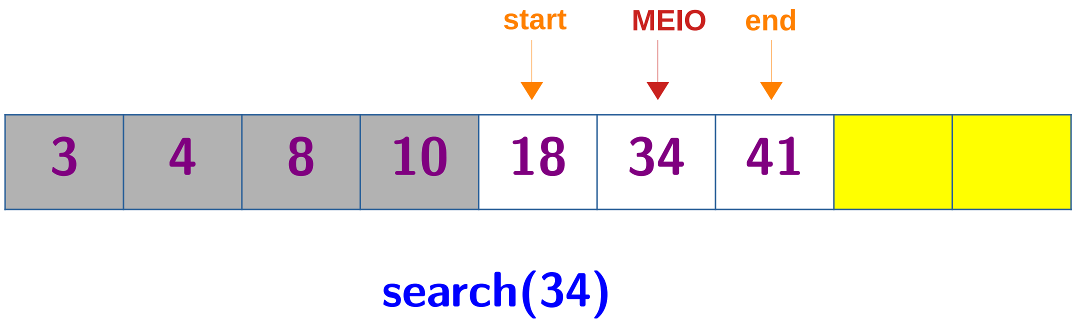
Busca Binária



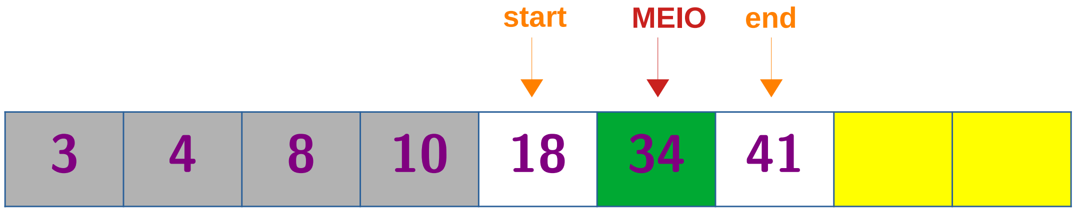
search(34)

Árvore de Busca Binária

Busca Binária



Árvore de Busca Binária



Busca Binária

search(34)

Árvore de Busca Binária

				start	MEIO	end		
3	4	8	10	18	34	41		

Busca Binária

search(34)

Espaço de busca

$$n \rightarrow n/2 \rightarrow n/4 \rightarrow \dots \rightarrow 1$$

$\log_2 n$ passos

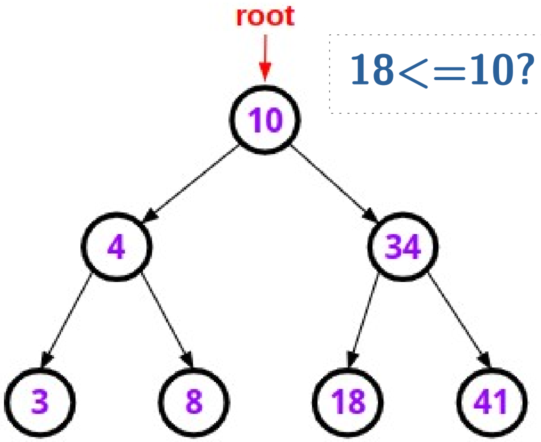
$n/2^k = 1$

$\rightarrow 2^k = n$

$\rightarrow k = \log_2 n$

Árvore de Busca Binária

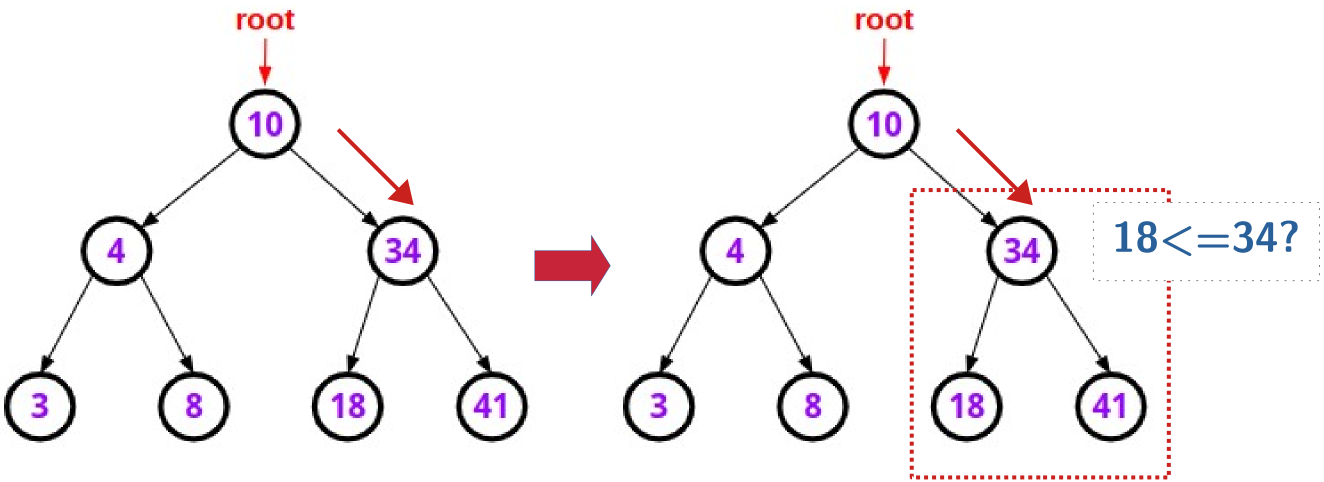
search(18)



18<=10?

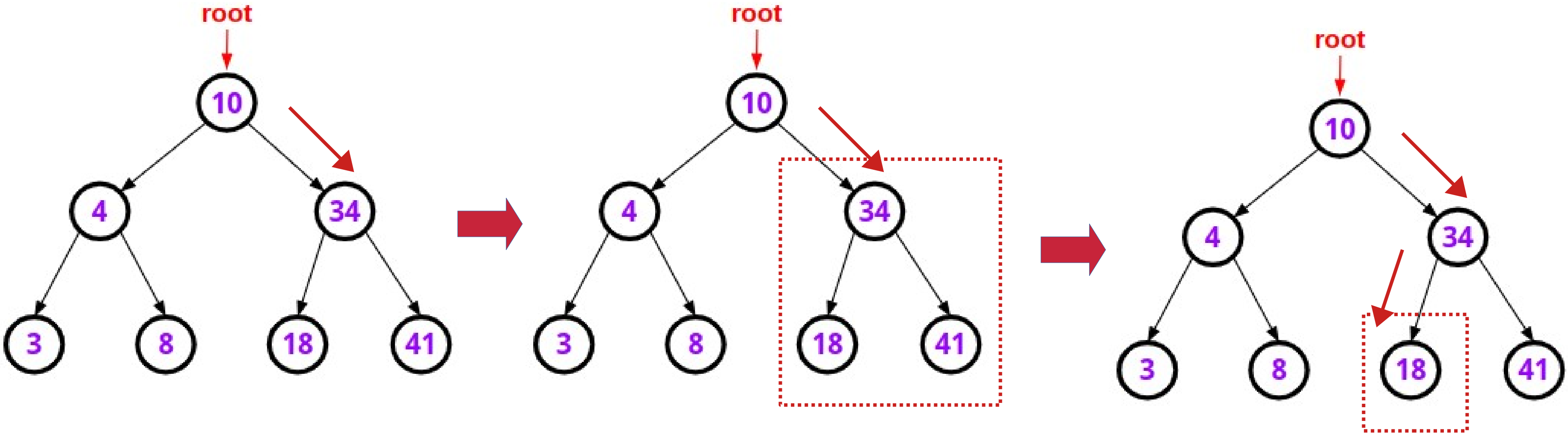
Árvore de Busca Binária

search(18)



Árvore de Busca Binária

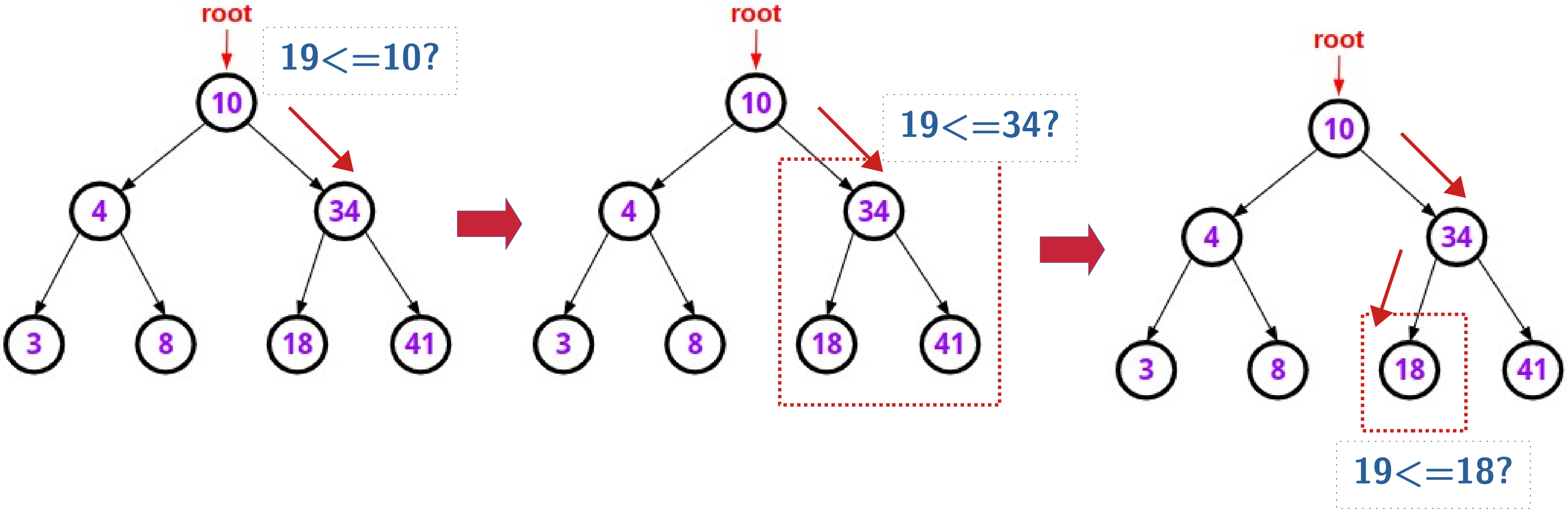
search(18)



$18 <= 18?$

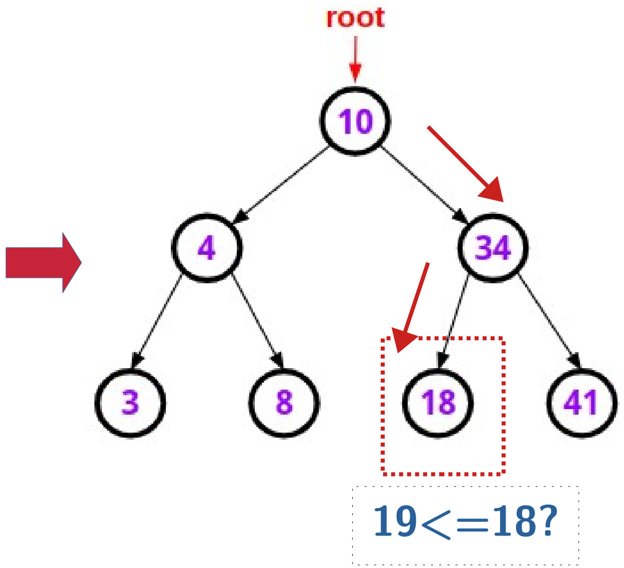
Árvore de Busca Binária

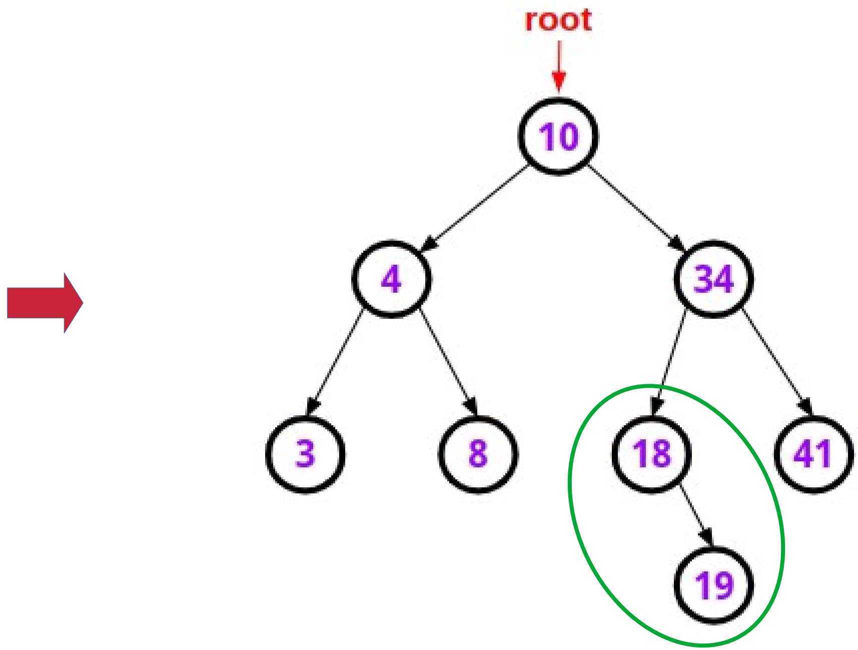
insert(19)



Árvore de Busca Binária

insert(19)





Árvore de Busca Binária

Árvore de Busca Binária

Árvore de Busca Binária

Árvore de Busca Binária

Árvore de Busca Binária

Árvore de Busca Binária