

ALGORITMOS EM GRAFOS

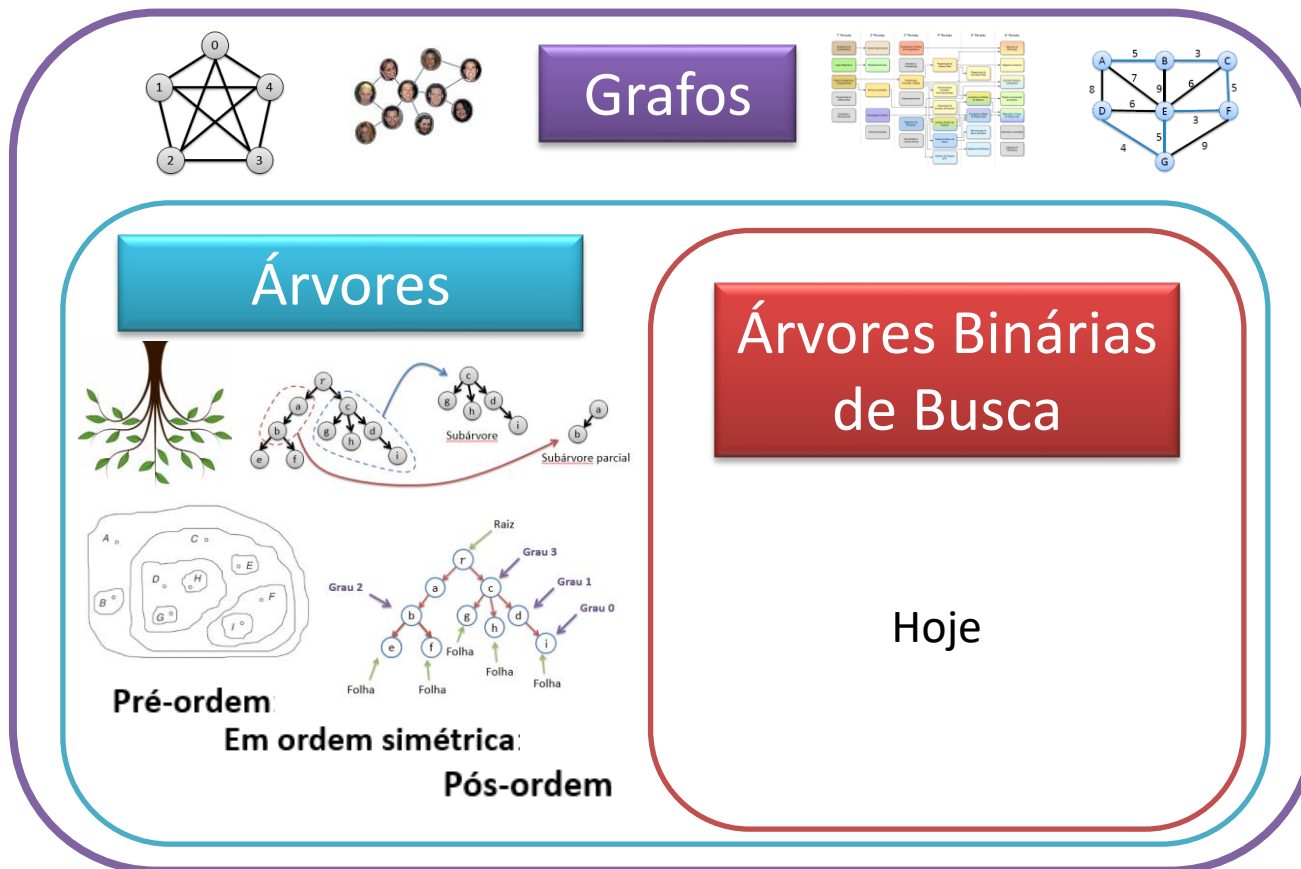
Bacharelado em Sistemas da Informação

Prof. Marco André Abud Kappel

Aula 7 – Árvores Binárias de Busca

Árvores Binárias de Busca

- Anteriormente:



Árvores Binárias de Busca

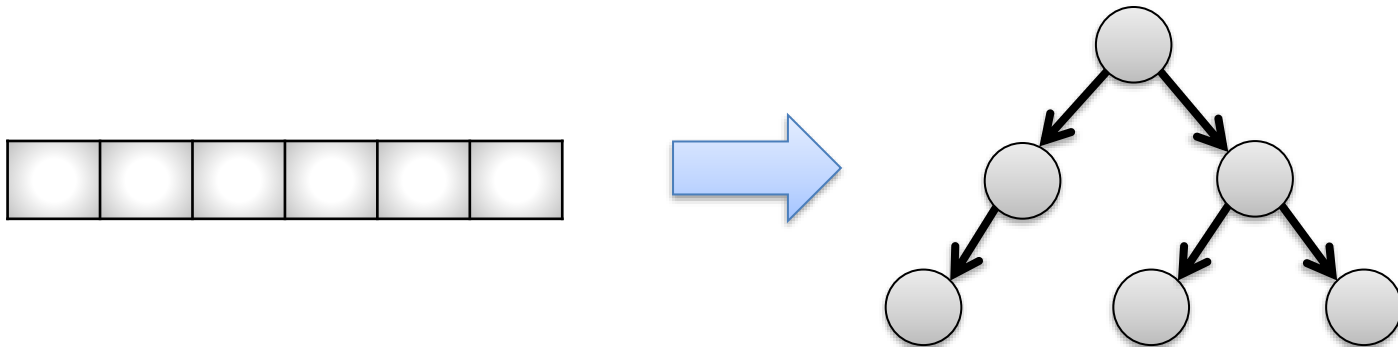
- **Introdução**

- Veremos, de agora em diante, estruturas de dados adequadas à solução de **problemas de busca**.
- Dado um **conjunto de elementos**, onde cada um é identificado por uma **chave**, o objetivo é **localizar** nesse conjunto o **elemento** correspondente a uma **chave específica** procurada.
- Anteriormente, já foram vistos **diversos métodos** diferentes para resolver este problema, como **busca linear**, **busca linear ordenada**, **busca binária**, etc.

Árvores Binárias de Busca

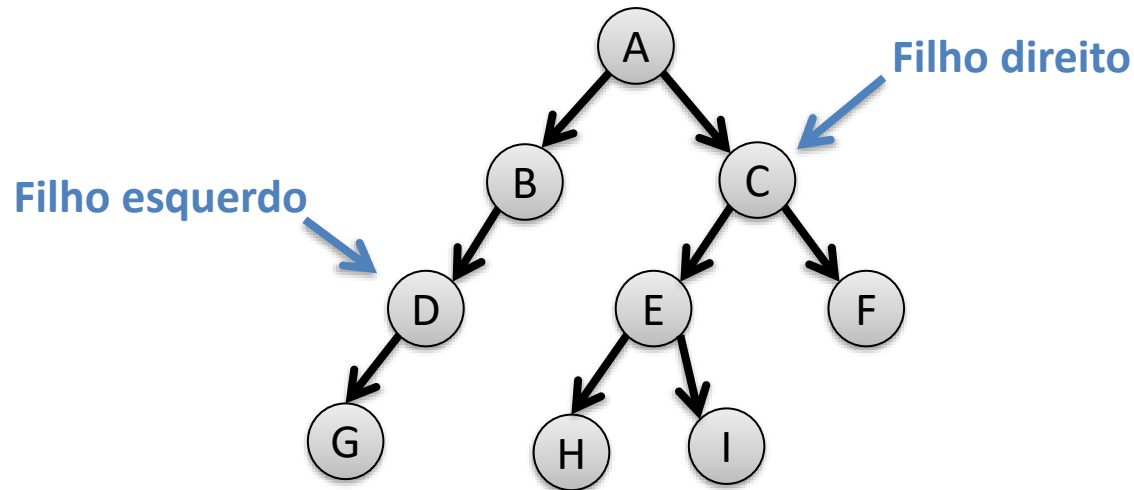
- **Introdução**

- Agora, veremos métodos de solução que empregam determinados tipos de **árvores** como estruturas nas quais se **processa a busca**.
- Ou seja, os **elementos** do conjunto são **previamente distribuídos** pelos nós de uma **árvore** de forma conveniente.



Árvores Binárias de Busca

- **Árvores Binárias**
 - Cada nó possui, no máximo, **dois filhos**: uma **subárvore esquerda** e/ou uma **subárvore direita**.



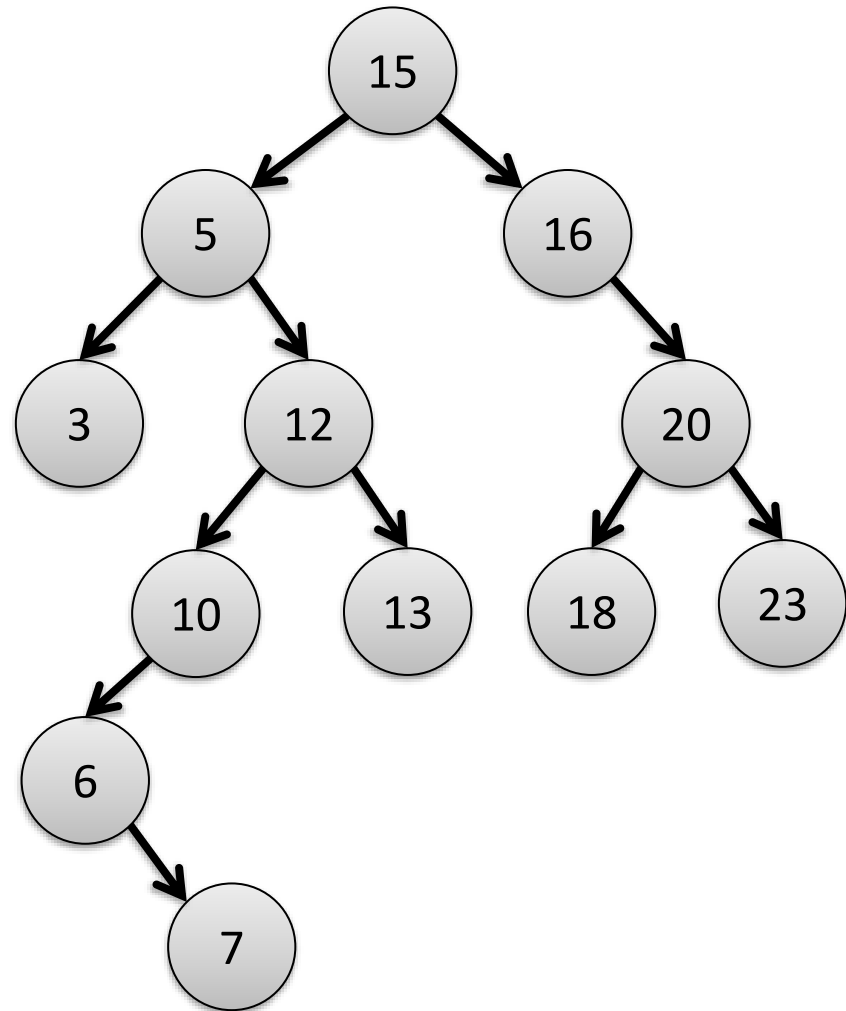
Árvores Binárias de Busca

- **Árvores Binárias de Busca**

- Uma **Árvore Binária de Busca** é uma árvore binária com as seguintes propriedades:
 - Cada **nó** possui uma **chave**.
 - As **chaves** dos elementos em qualquer **subárvore esquerda** (se houver) são **menores** do que o valor em seu nó **pai**.
 - As **chaves** dos elementos em qualquer **subárvore direita** (se houver) são **maiores** do que o valor em seu nó **pai**.
 - As **subárvores esquerda e direita** são **árvores binárias de busca**.

Árvores Binárias de Busca

- **Árvores Binárias de Busca**
 - A **subárvore da esquerda** possui somente nós com **valores menores** que o da sua raiz.
 - A **subárvore da direita** possui somente nós com **valores maiores** que o da sua raiz.



Árvores Binárias de Busca

- **Árvores Binárias de Busca**

- Existem **três operações básicas** para as **Árvores Binárias de Busca**:
 - **Buscar** um elemento
 - **Inserir** um elemento
 - **Remover** um elemento.
- Todas essas operações devem ser feitas levando em consideração a **estrutura da árvore**.
- **Inserir** ou **remover** um elemento **não** deve fazer a árvore **deixar de ser Binária de Busca**.

Árvores Binárias de Busca

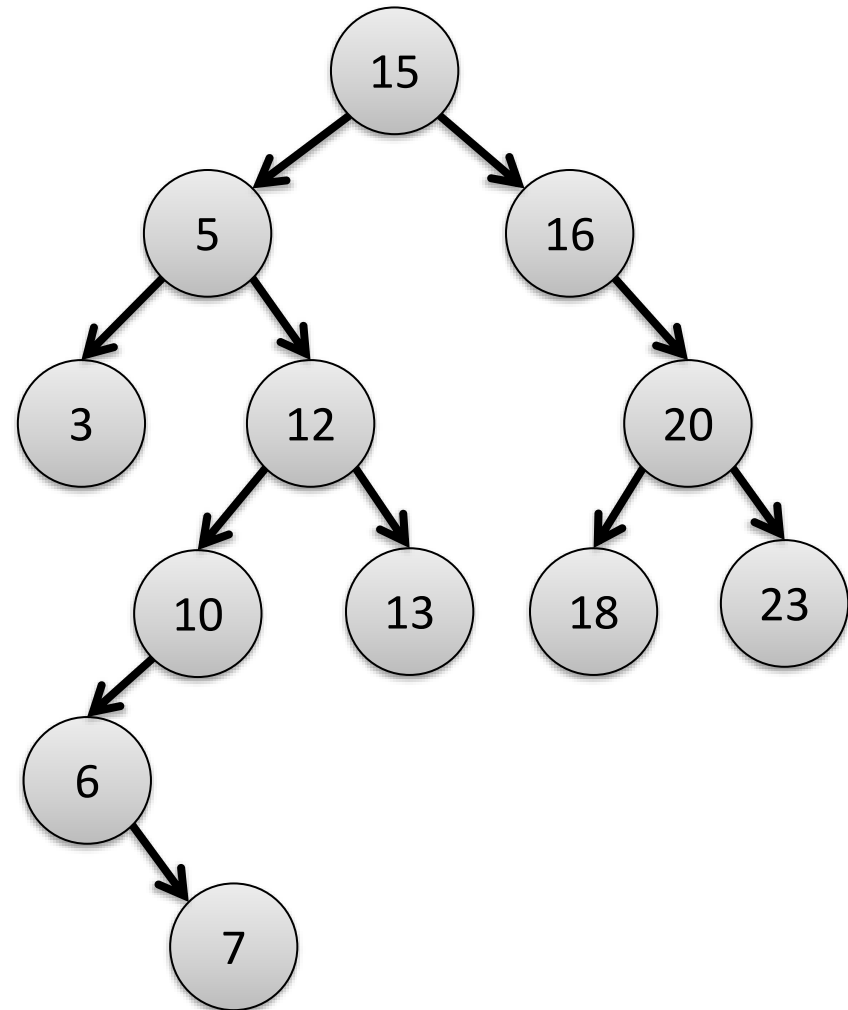
- **Buscar um elemento**
 - É a **principal função** da Árvore Binária de Busca.
 - A solução **natural** é **recursiva**.
 - Algoritmo:
 1. Comece a **busca** pelo **nó raiz**.
 2. Se a **árvore** for **vazia**, então retorne **NULL**.
 3. Se a **chave** for **igual**, retorne o **ponteiro** para o **elemento**.
 4. Se a **chave** for **menor**, aplique a **busca** na **subárvore esquerda**.
 5. Se a **chave** for **maior**, aplique a **busca** na **subárvore direita**.

Árvores Binárias de Busca

- **Buscar um elemento**

- Exemplo:

- Buscar 13:



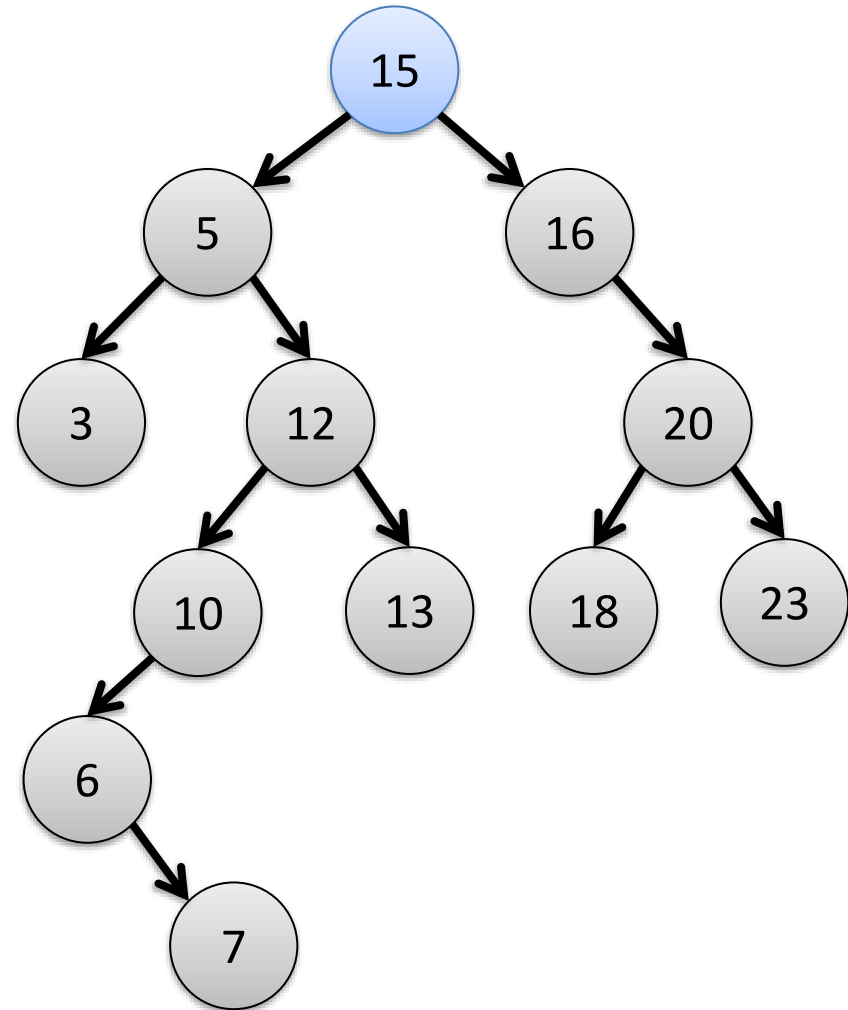
Árvores Binárias de Busca

- **Buscar um elemento**

- Exemplo:

- Buscar 13:

15 = 13?



Árvores Binárias de Busca

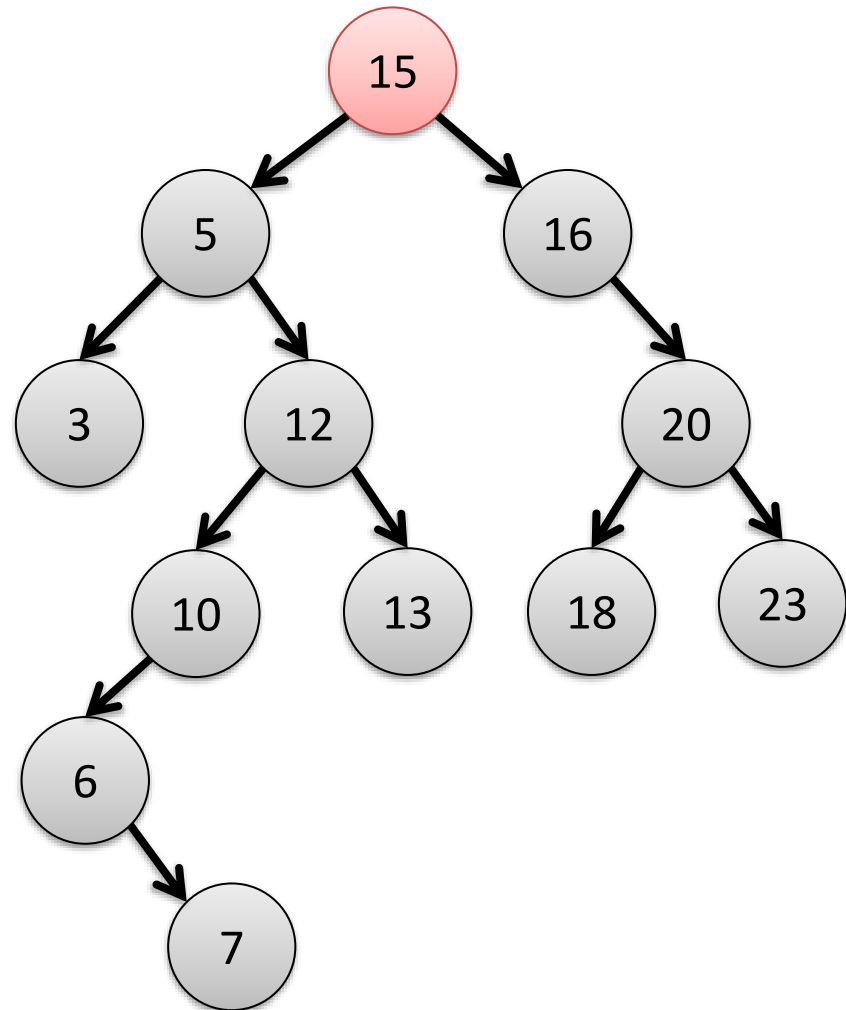
- **Buscar um elemento**

- Exemplo:

- Buscar 13:

15 = 13? Não!

13 < 15?



Árvores Binárias de Busca

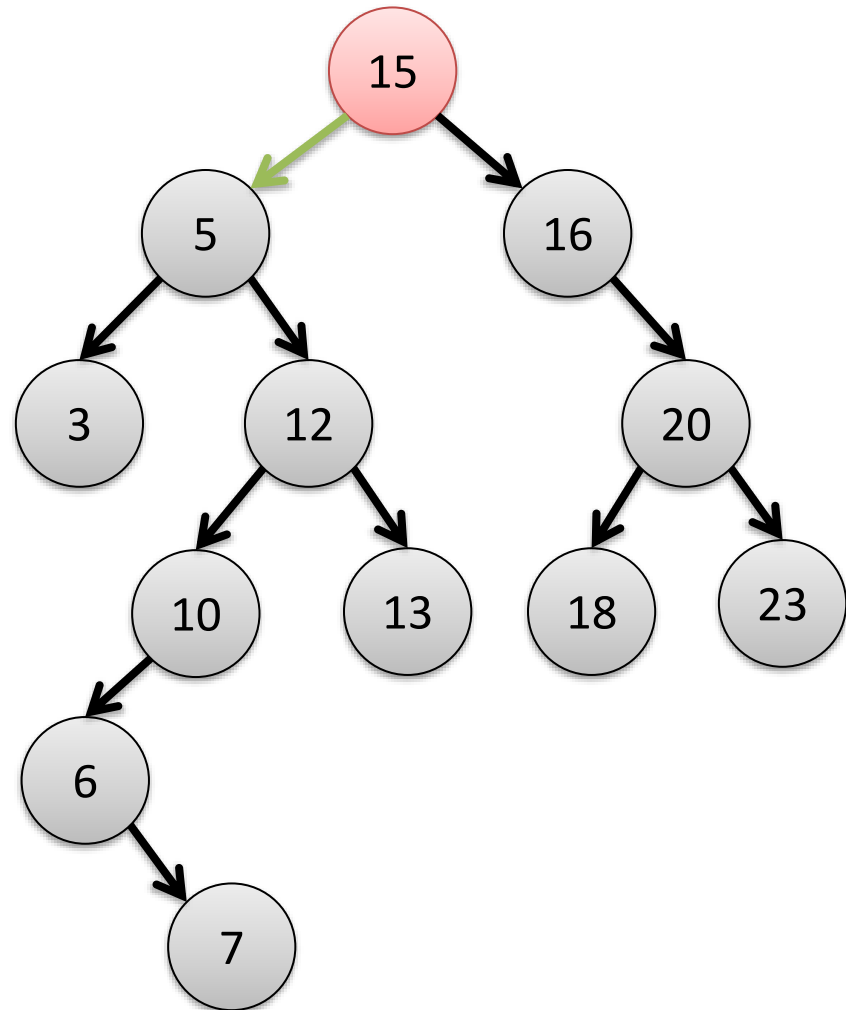
- **Buscar um elemento**

- Exemplo:

- Buscar 13:

15 = 13? Não!

13 < 15? Sim! Busca na subárvore esquerda.



Árvores Binárias de Busca

- **Buscar um elemento**

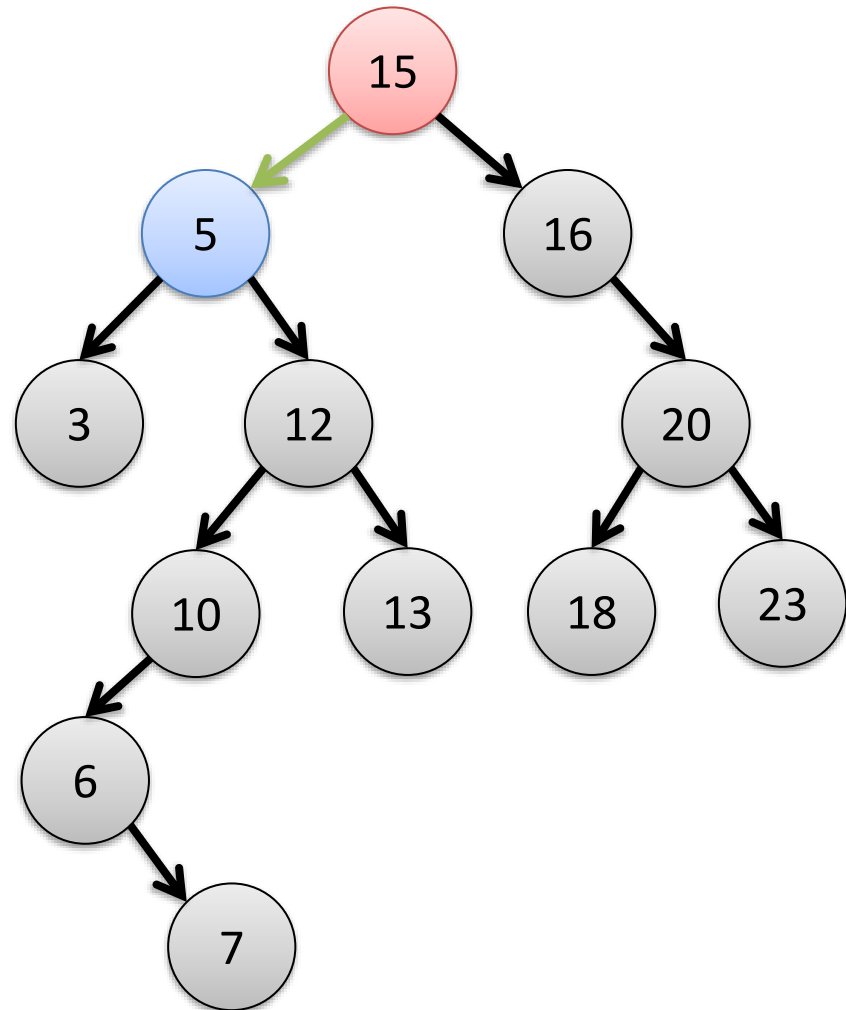
- Exemplo:

- Buscar 13:

15 = 13? Não!

13 < 15? Sim! Busca na subárvore esquerda.

5 = 13?



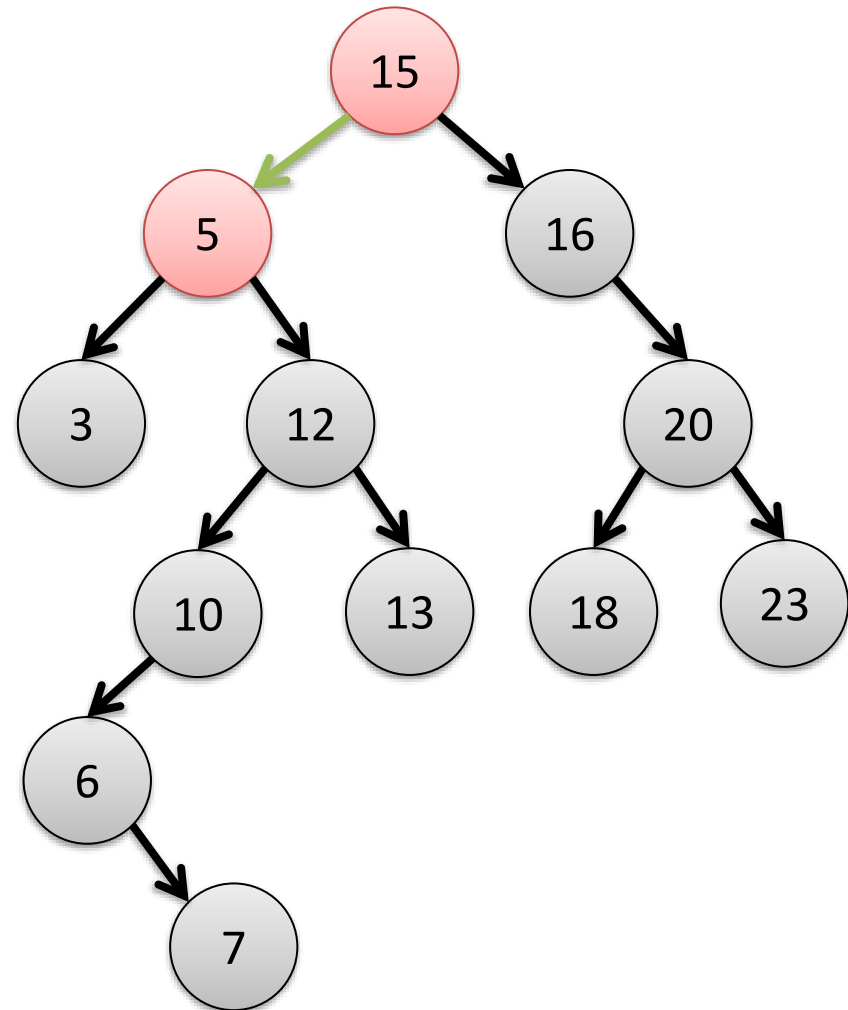
Árvores Binárias de Busca

- **Buscar um elemento**

- Exemplo:

- Buscar 13:

15 = 13? Não!
13 < 15? Sim! Busca na subárvore esquerda.
5 = 13? Não!
13 < 5?



Árvores Binárias de Busca

- **Buscar um elemento**

- Exemplo:

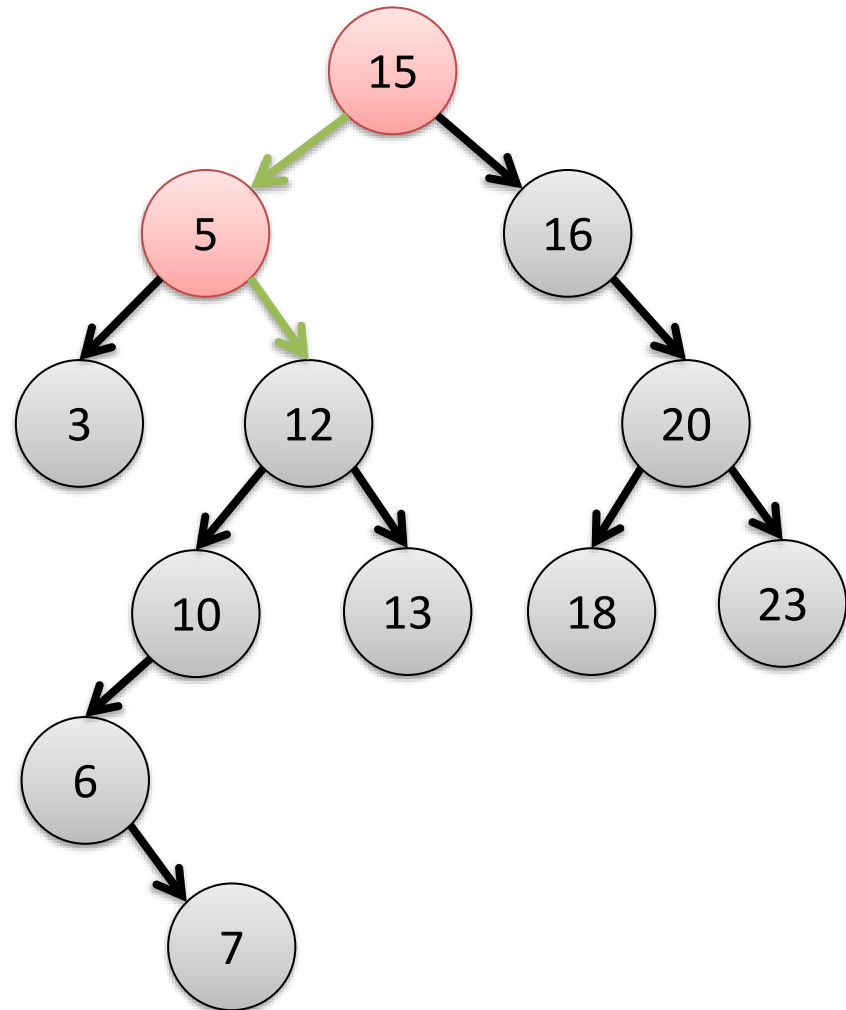
- Buscar 13:

15 = 13? Não!

13 < 15? Sim! Busca na subárvore esquerda.

5 = 13? Não!

13 < 5? Não! Busca na subárvore direita.



Árvores Binárias de Busca

- **Buscar um elemento**

- Exemplo:

- Buscar 13:

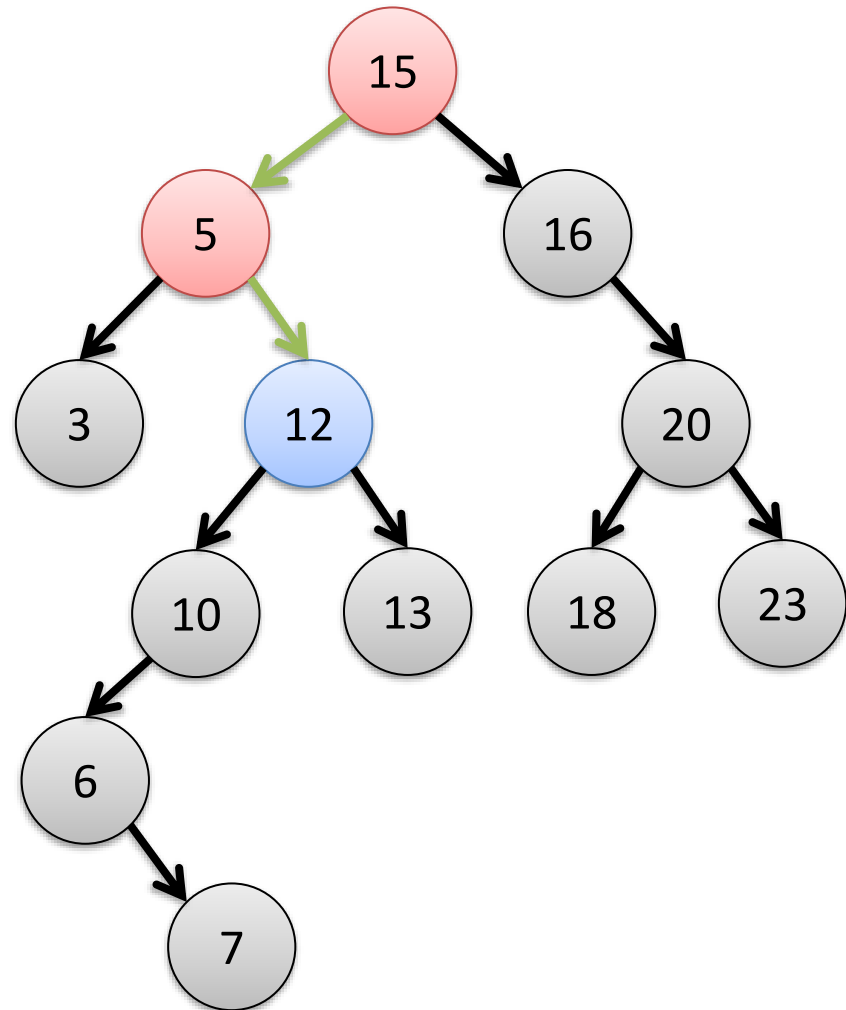
15 = 13? Não!

13 < 15? Sim! Busca na subárvore esquerda.

5 = 13? Não!

13 < 5? Não! Busca na subárvore direita.

12 = 13?



Árvores Binárias de Busca

- **Buscar um elemento**

- Exemplo:

- Buscar 13:

15 = 13? Não!

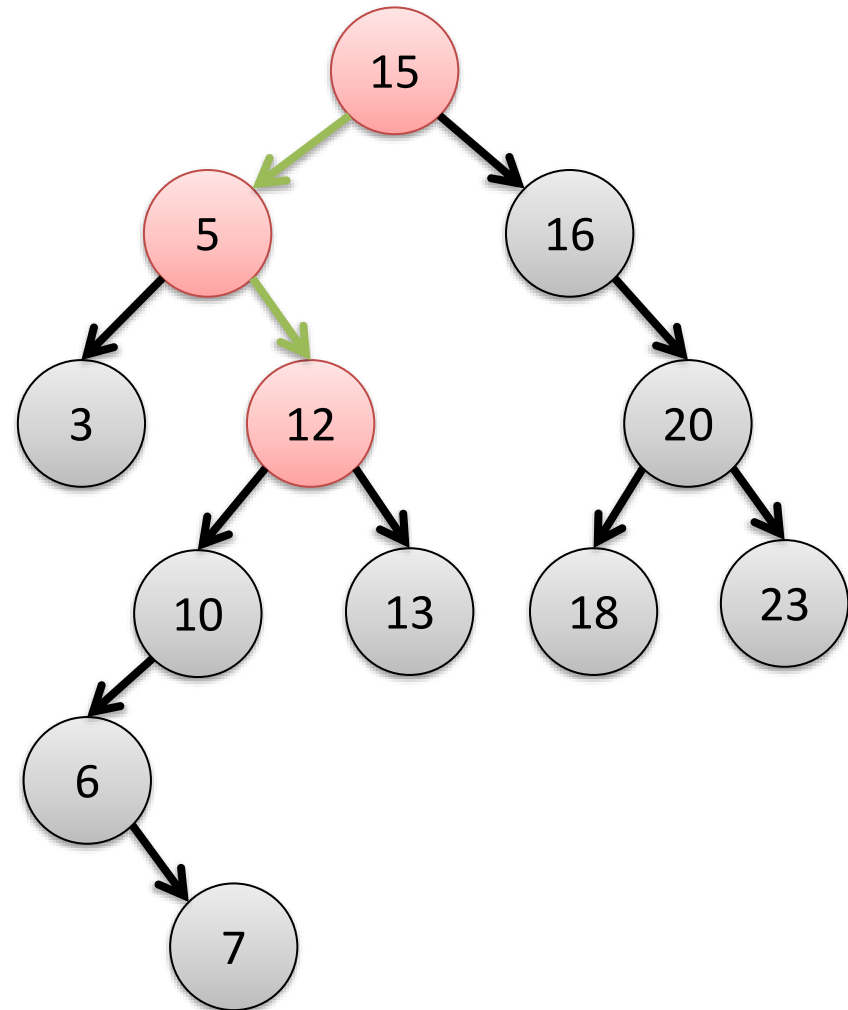
13 < 15? Sim! Busca na subárvore esquerda.

5 = 13? Não!

13 < 5? Não! Busca na subárvore direita.

12 = 13? Não!

13 < 12?



Árvores Binárias de Busca

- **Buscar um elemento**

- Exemplo:

- Buscar 13:

15 = 13? Não!

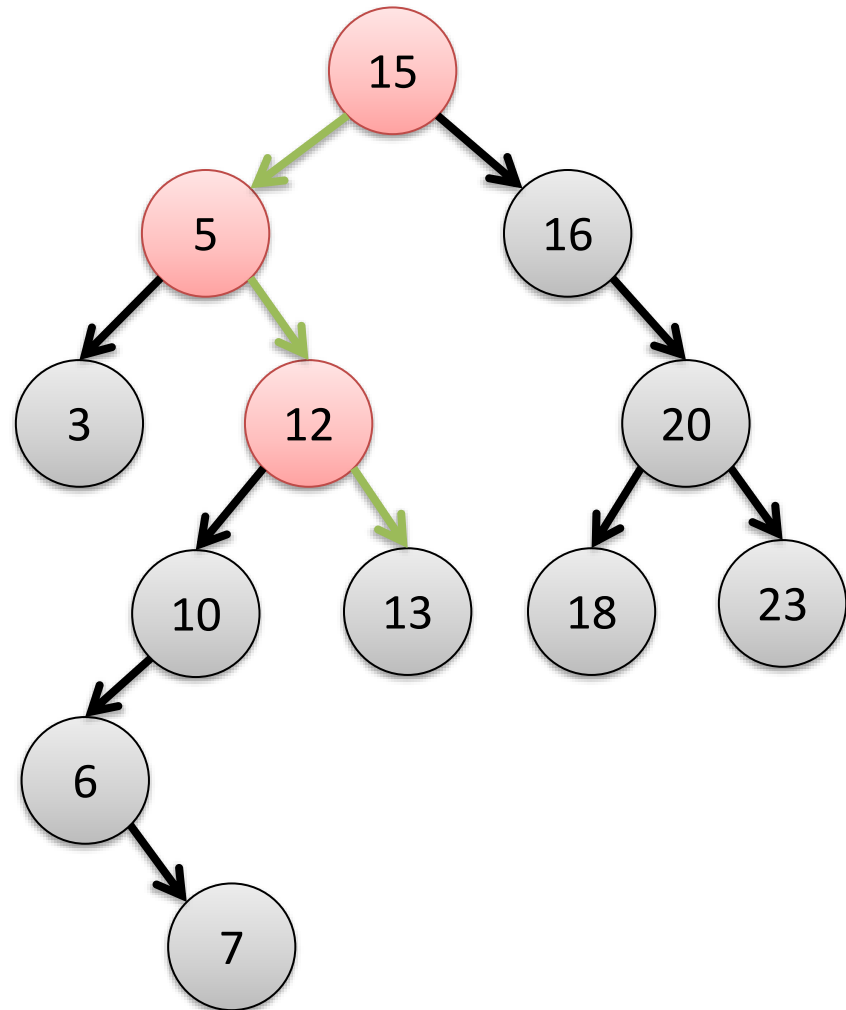
13 < 15? Sim! Busca na subárvore esquerda.

5 = 13? Não!

13 < 5? Não! Busca na subárvore direita.

12 = 13? Não!

13 < 12? Não! Busca na subárvore direita.



Árvores Binárias de Busca

- **Buscar um elemento**

- Exemplo:

- Buscar 13:

15 = 13? Não!

13 < 15? Sim! Busca na subárvore esquerda.

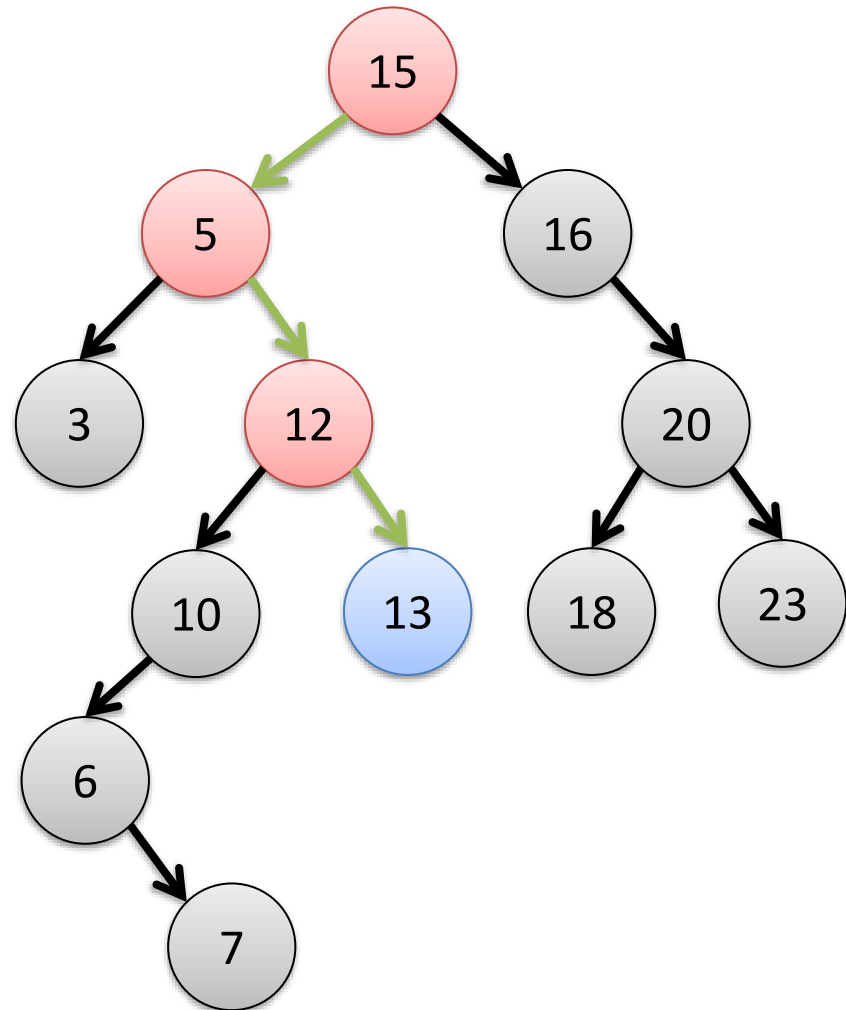
5 = 13? Não!

13 < 5? Não! Busca na subárvore direita.

12 = 13? Não!

13 < 12? Não! Busca na subárvore direita.

13 = 13?



Árvores Binárias de Busca

- **Buscar um elemento**

- Exemplo:

- Buscar 13:

15 = 13? Não!

13 < 15? Sim! Busca na subárvore esquerda.

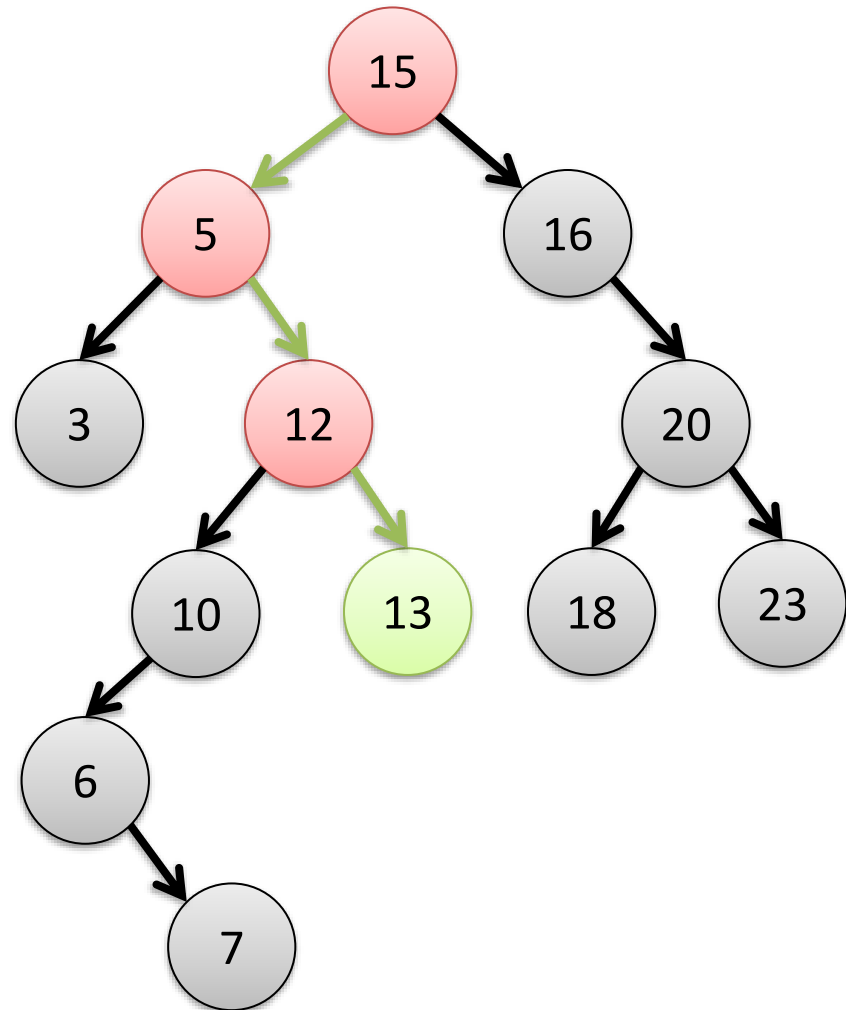
5 = 13? Não!

13 < 5? Não! Busca na subárvore direita.

12 = 13? Não!

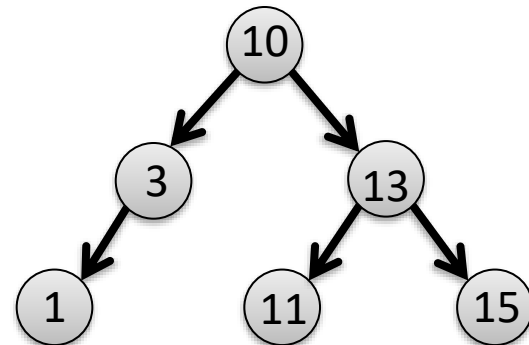
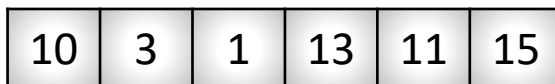
13 < 12? Não! Busca na subárvore direita.

13 = 13? Sim! Retorna um ponteiro para este elemento!



Árvores Binárias de Busca

- Inserir um elemento
 - O processo de **construção** de uma ABB é, **basicamente**, uma **sequência** de **inclusões** de elementos.
 - É importante que, **ao final da inserção**, a árvore **continue** sendo uma **ABB**.
 - A solução natural é recursiva.



- **Inserir um elemento**

- Algoritmo:

1. Comece pelo **nó raiz**.
2. Se o nó for NULL, **crie um nó** com a **chave dada** e retorne o **endereço** dele.
3. Se a **chave** dada for **maior** que a **chave corrente**:
 1. Se o nó tiver uma **subárvore direita**, então **insira o elemento** na subárvore direita.
 2. Senão, **crie um nó** com a **chave dada** e este será o **filho à direita** do nó corrente.
4. Se a **chave** dada for **menor** que a **chave corrente**:
 1. Se o nó tiver uma **subárvore à esquerda**, então **insira o elemento** na subárvore esquerda.
 2. Senão, **crie um nó** com a chave dada e este será o **filho à esquerda** do nó corrente.
5. Se a chave for **igual**, troque a informação associada à chave (na **árvore de inteiros**, não faça nada).

Árvores Binárias de Busca

- **Inserir um elemento**

- Exemplo:
- Inserir os seguintes elementos nesta ordem:

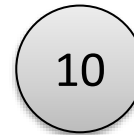
10, 15, 5, 8, 12, 16

Árvores Binárias de Busca

- **Inserir um elemento**

- Exemplo:
- Inserir os seguintes elementos nesta ordem:

10, 15, 5, 8, 12, 16

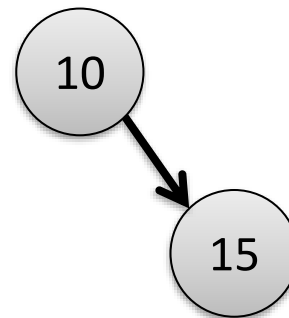


Árvores Binárias de Busca

- **Inserir um elemento**

- Exemplo:
- Inserir os seguintes elementos nesta ordem:

10, 15, 5, 8, 12, 16



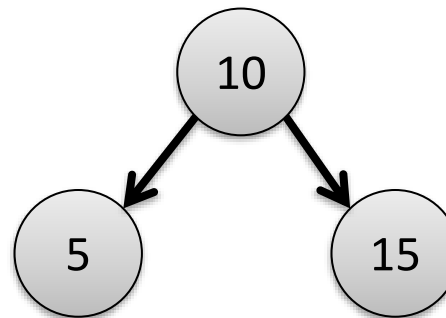
Árvores Binárias de Busca

- **Inserir um elemento**

- Exemplo:

- Inserir os seguintes elementos nesta ordem:

10, 15, 5, 8, 12, 16



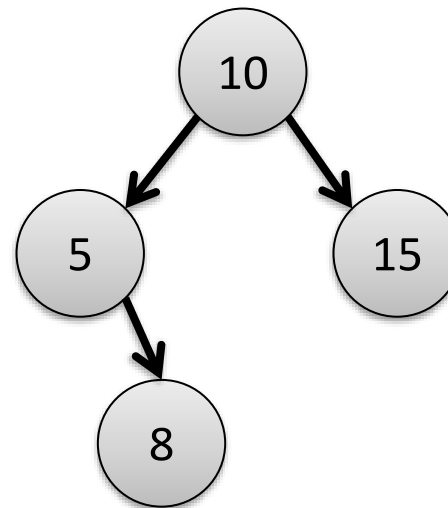
Árvores Binárias de Busca

- **Inserir um elemento**

- Exemplo:

- Inserir os seguintes elementos nesta ordem:

10, 15, 5, 8, 12, 16



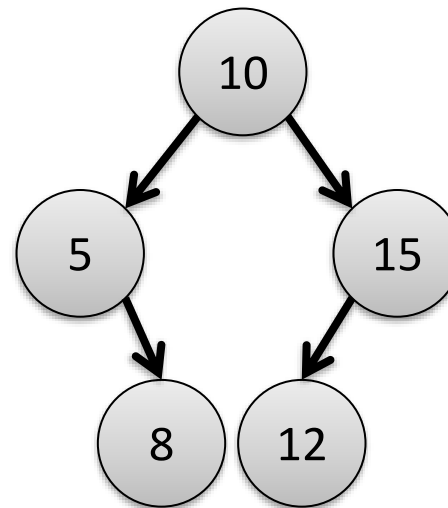
Árvores Binárias de Busca

- **Inserir um elemento**

- Exemplo:

- Inserir os seguintes elementos nesta ordem:

10, 15, 5, 8, 12, 16

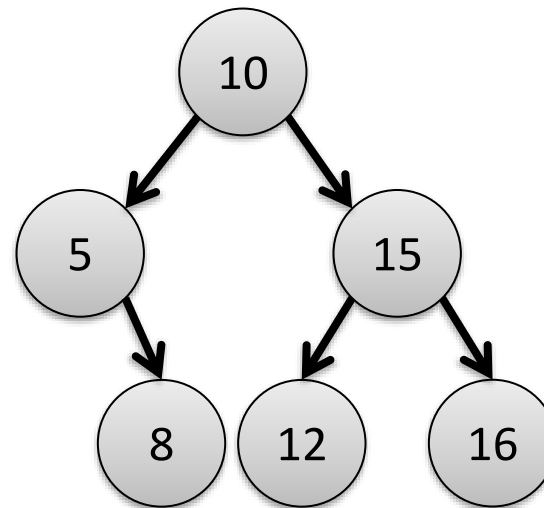


Árvores Binárias de Busca

- **Inserir um elemento**

- Exemplo:
- Inserir os seguintes elementos nesta ordem:

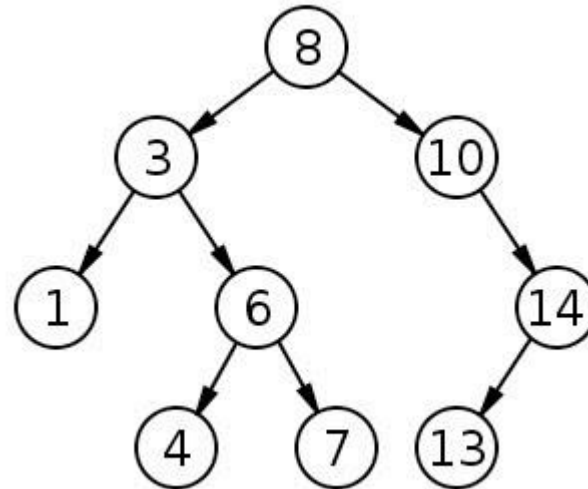
10, 15, 5, 8, 12, 16



Árvores Binárias de Busca

- **Exercício 1:**

a) A seguinte árvore é uma ABB?

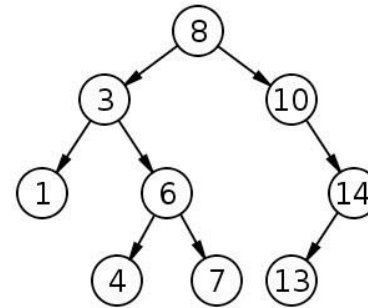


b) Insira os seguintes elementos em uma ABB, nesta ordem: 15, 20, 10, 25, 5, 18, 30, 22, 19, 16, 17.

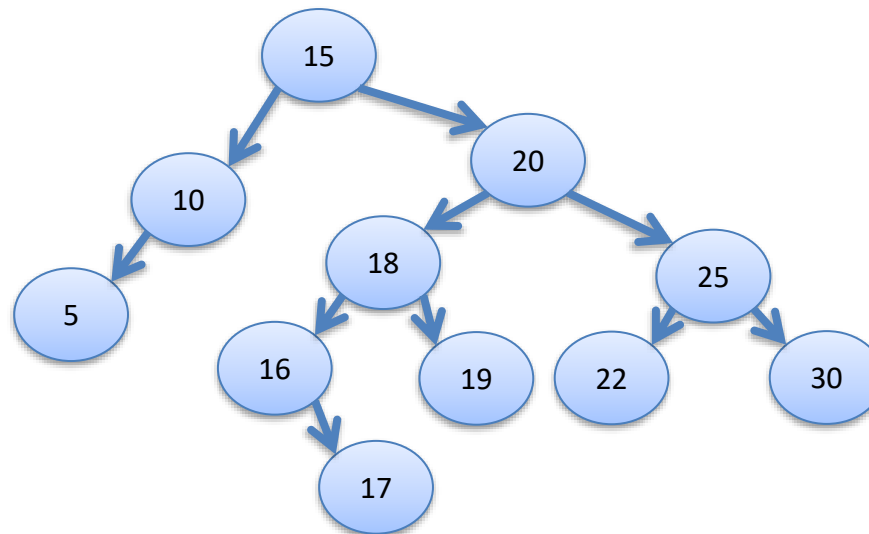
c) Ordene os elementos do item anterior e faça a inserção deles numa ABB.

- **Exercício 1:**

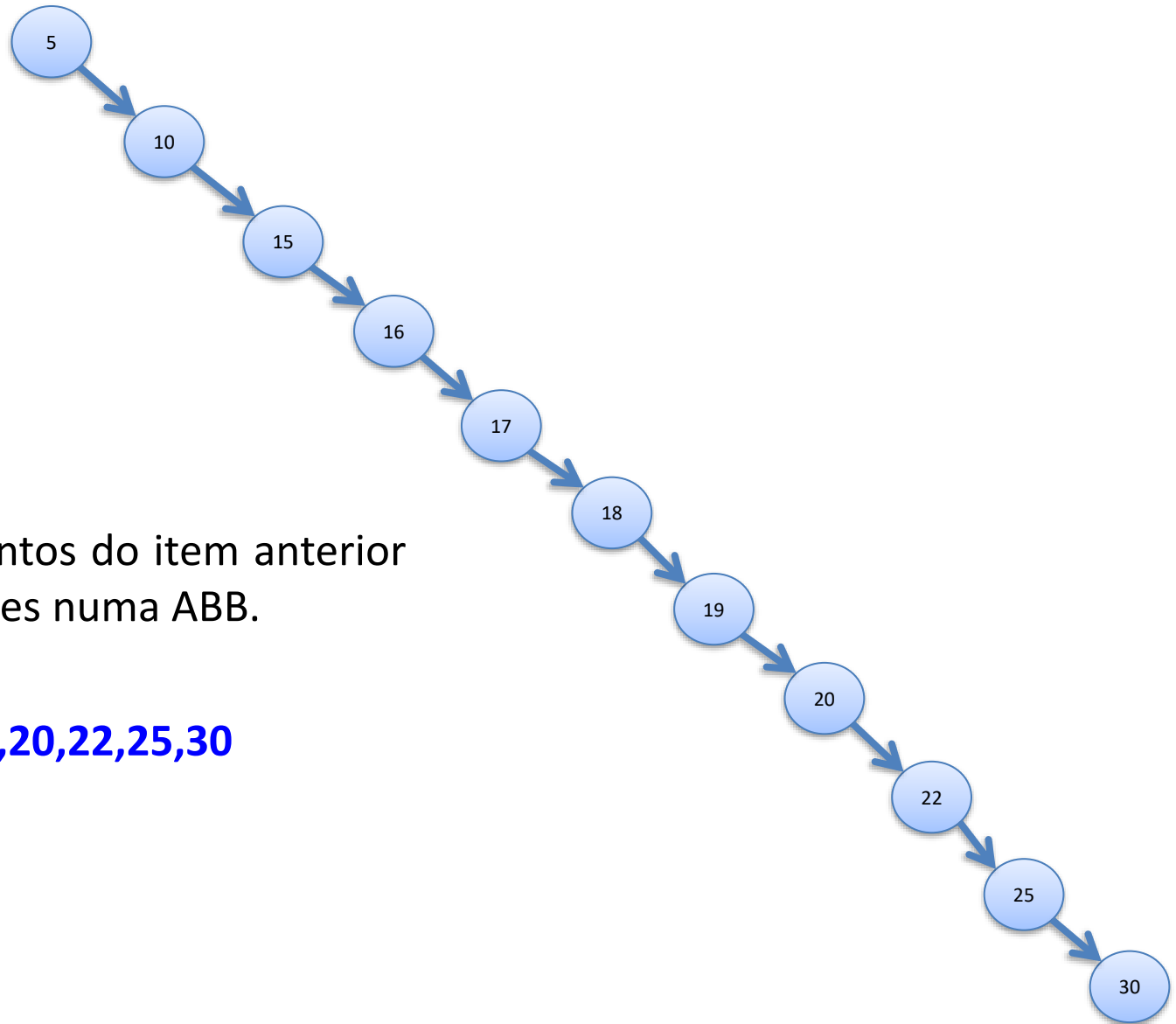
a) A seguinte árvore é uma ABB? **Sim**



b) Insira os seguintes elementos em uma ABB, nesta ordem: 15, 20, 10, 25, 5, 18, 30, 22, 19, 16, 17.



- Exercício 1:



c) Ordene os elementos do item anterior e faça a inserção deles numa ABB.

5,10,15,16,17,18,19,20,22,25,30

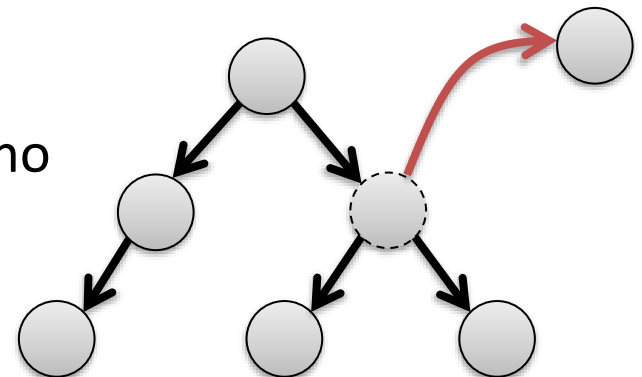
Árvores Binárias de Busca

- **Excluir um elemento**

- A operação de **exclusão** pode ser a mais **complicada** dentre as operações nas árvores binárias de busca.

- **Três casos** precisam ser verificados:

1. O nó a ser excluído é uma folha
2. O nó a ser excluído tem apenas 1 filho
3. O nó a ser excluído tem dois filhos



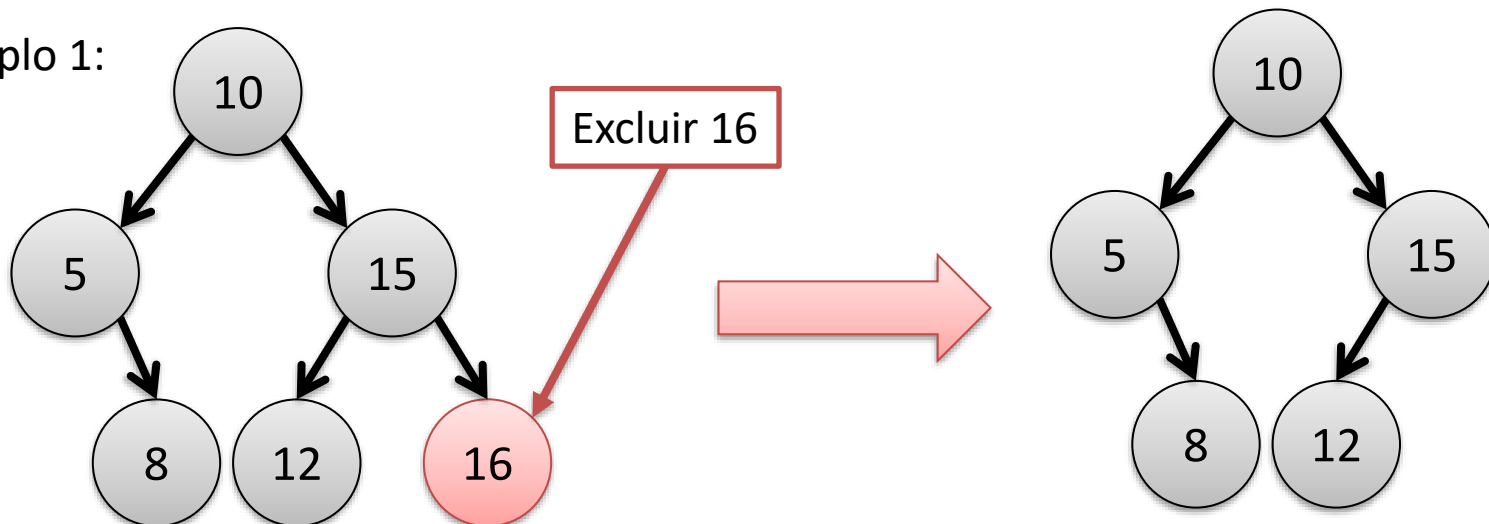
Árvores Binárias de Busca

- **Excluir um elemento**

- Caso 1: O nó a ser excluído é uma folha

- Simplesmente elimina o nó e o pai passa a apontar para NULL.

Exemplo 1:

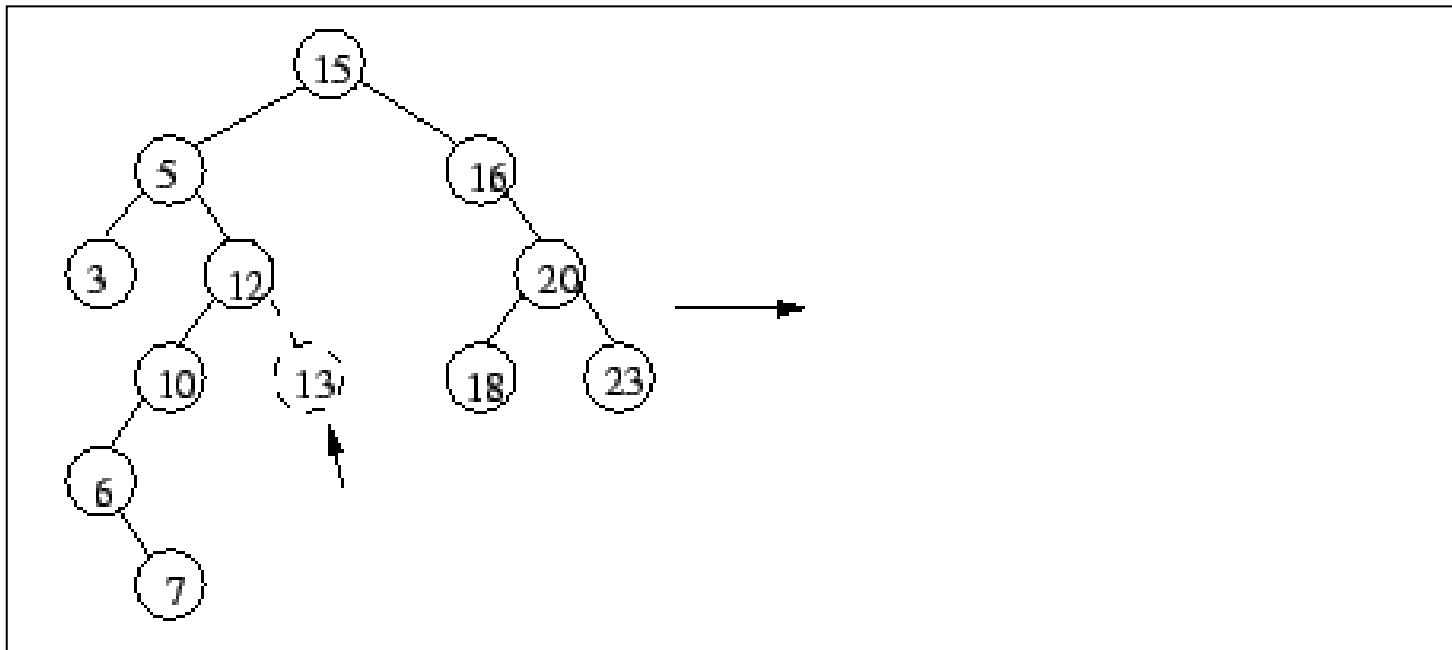


Árvores Binárias de Busca

- **Excluir um elemento**

- Caso 1: O nó a ser excluído é uma folha

Exemplo 2:

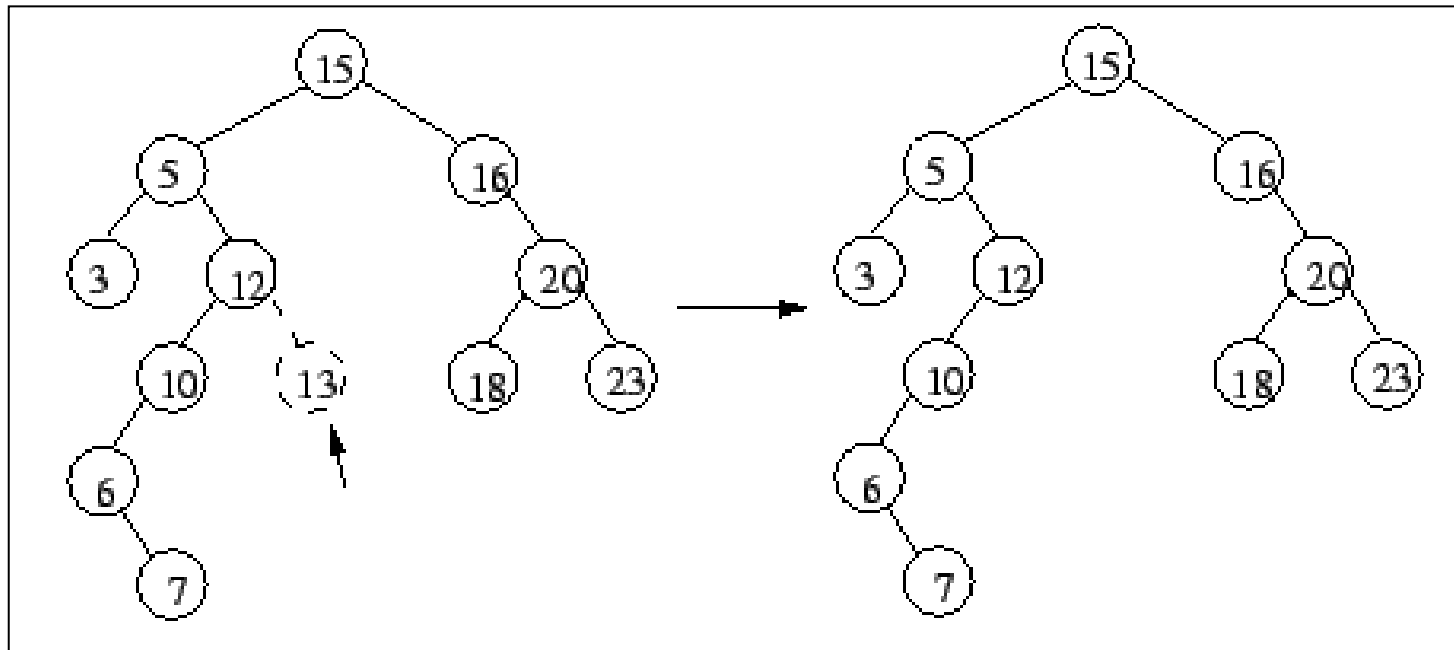


Árvores Binárias de Busca

- **Excluir um elemento**

- Caso 1: O nó a ser excluído é uma folha

Exemplo 2:



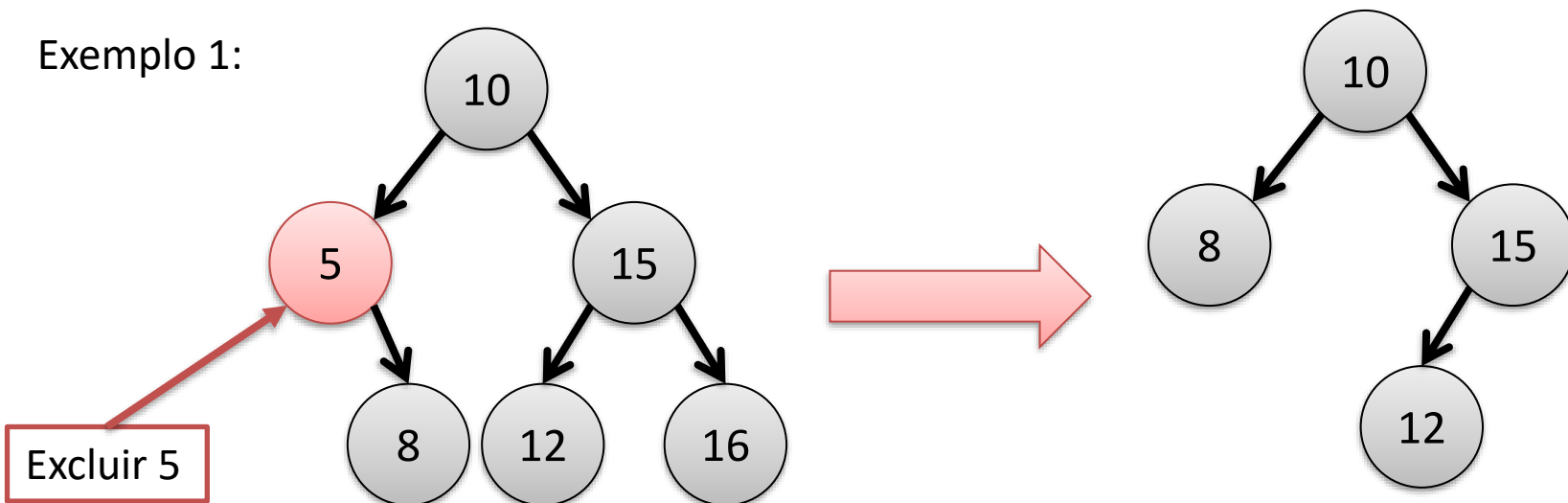
Árvores Binárias de Busca

- **Excluir um elemento**

- Caso 2: O nó a ser excluído tem apenas 1 filho

- A subárvore é **promovida**, ou seja, exclui o nó e faz o pai apontar para o filho do excluído.

Exemplo 1:

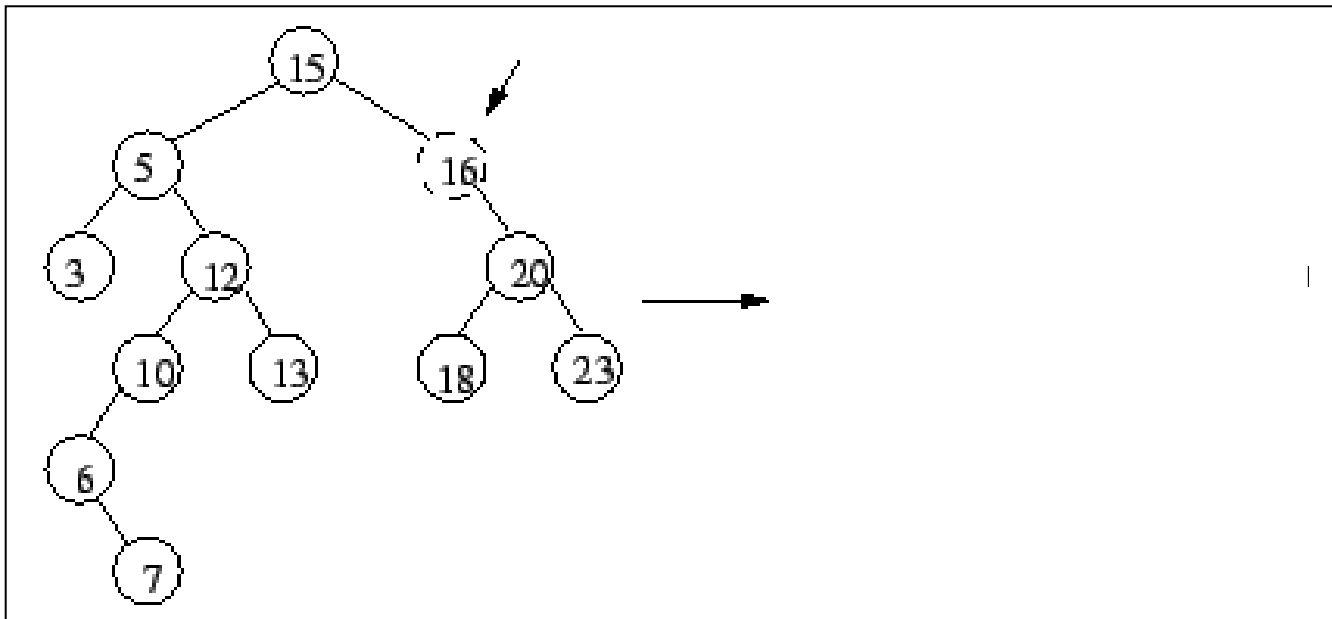


Árvores Binárias de Busca

- **Excluir um elemento**

➤ Caso 2: O nó a ser excluído tem apenas 1 filho

Exemplo 2:

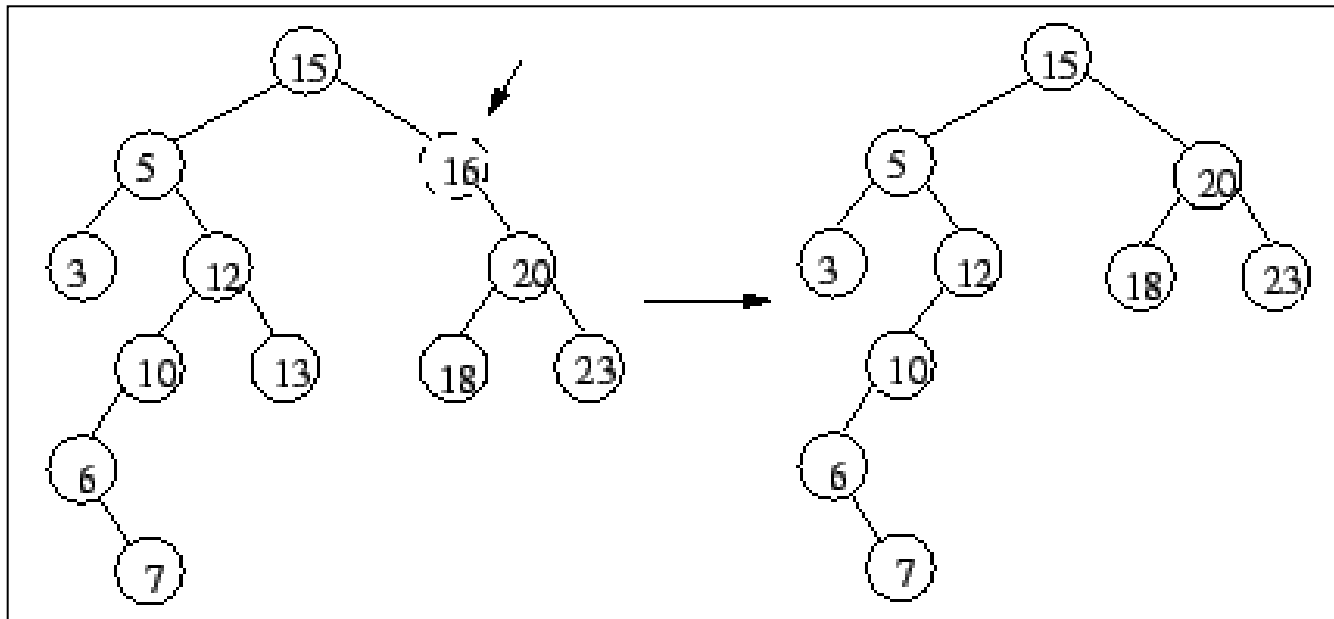


Árvores Binárias de Busca

- **Excluir um elemento**

- Caso 2: O nó a ser excluído tem apenas 1 filho

Exemplo 2:



Árvores Binárias de Busca

- **Excluir um elemento**

- Caso 3: O nó a ser excluído tem 2 filhos

- Algoritmo tem **três passos**:

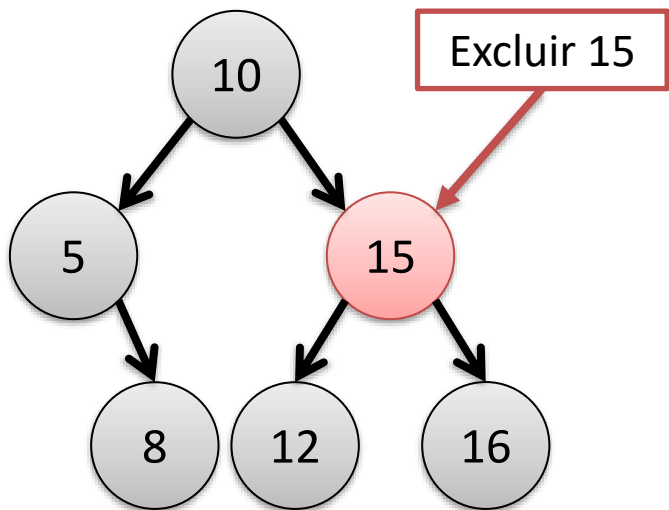
1. Buscar o **menor** elemento da **subárvore direita** do nó e **guardar** o seu valor.
2. **Substituir** o **valor** do nó a ser excluído inicialmente pelo **valor** do nó encontrado.
3. Aplicar, na **subárvore direita**, a **exclusão** do nó **encontrado** no passo anterior.

Árvores Binárias de Busca

- **Excluir um elemento**

- Caso 3: O nó a ser excluído tem 2 filhos

- Exemplo 1:

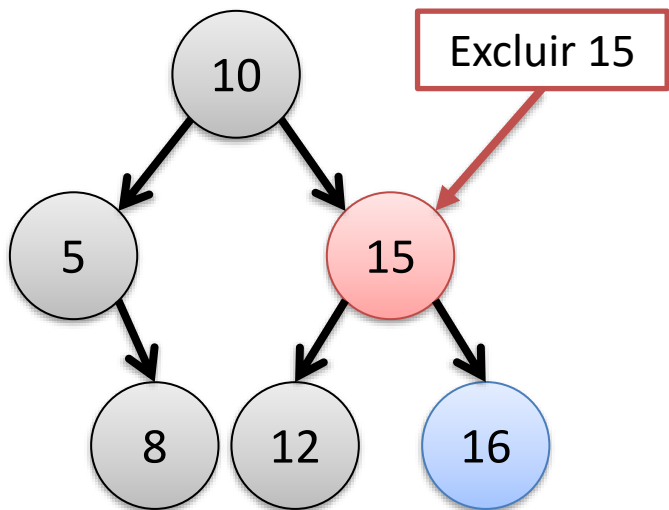


Árvores Binárias de Busca

- **Excluir um elemento**

- Caso 3: O nó a ser excluído tem 2 filhos

- Exemplo 1:



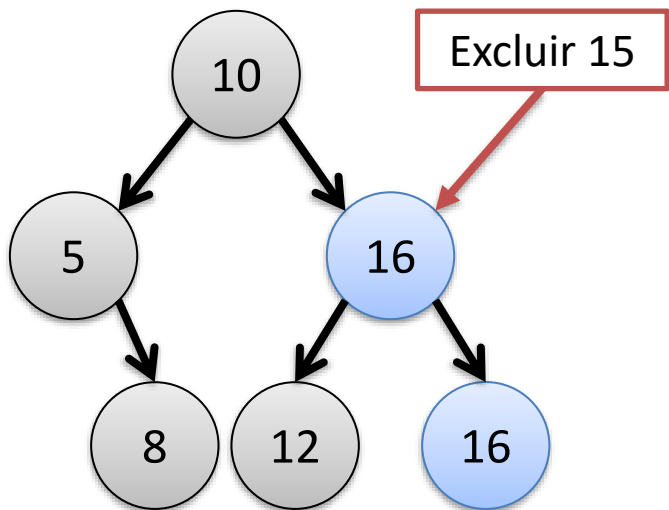
Passo 1: Encontrou o nó 16

Árvores Binárias de Busca

- **Excluir um elemento**

- Caso 3: O nó a ser excluído tem 2 filhos

- Exemplo 1:



Passo 1: Encontrou o nó 16

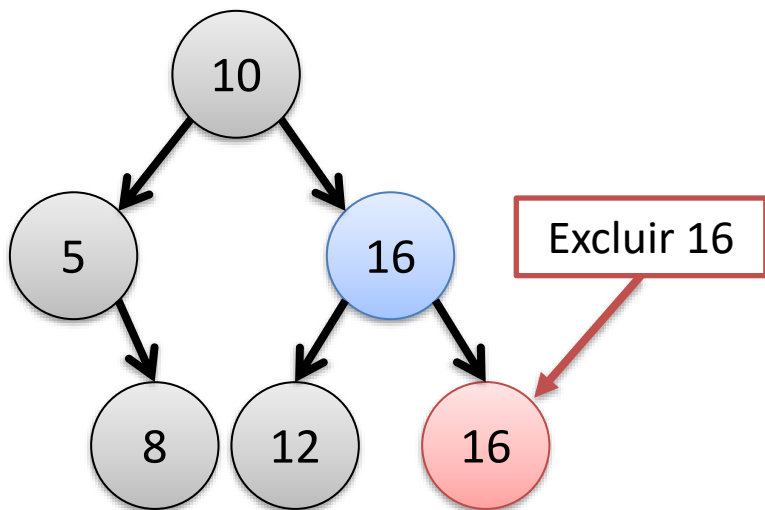
Passo 2: Copiar valor

Árvores Binárias de Busca

- **Excluir um elemento**

- Caso 3: O nó a ser excluído tem 2 filhos

- Exemplo 1:



Passo 1: Encontrou o nó 16

Passo 2: Copiar valor

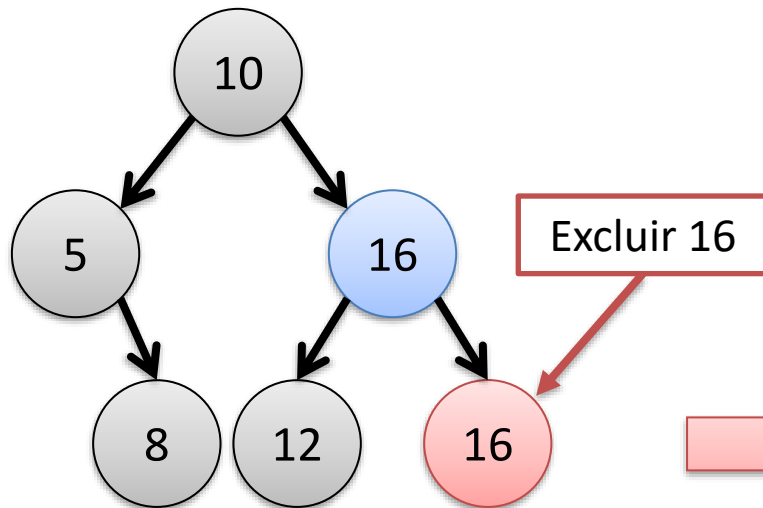
Passo 3: Excluir nó 16 da subárvore direita

Árvores Binárias de Busca

- **Excluir um elemento**

➤ Caso 3: O nó a ser excluído tem 2 filhos

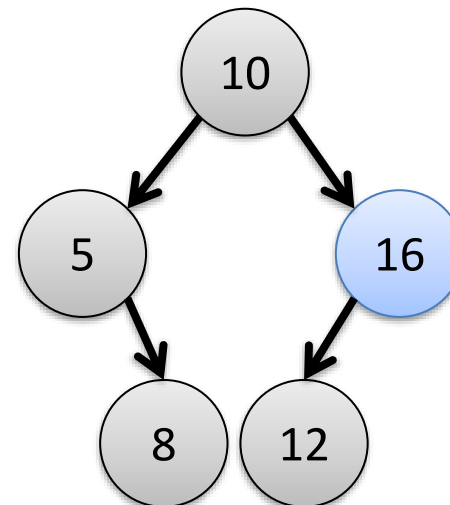
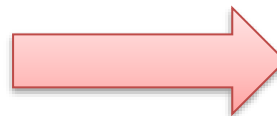
— Exemplo 1:



Passo 1: Encontrou o nó 16

Passo 2: Copiar valor

Passo 3: Excluir nó 16 da subárvore direita



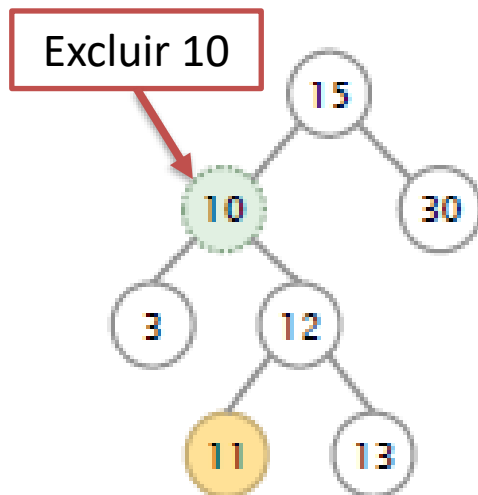
Árvores Binárias de Busca

- **Excluir um elemento**

- Caso 3: O nó a ser excluído tem 2 filhos

- Exemplo 2:

Passo 1: Encontrou o nó 11



Árvores Binárias de Busca

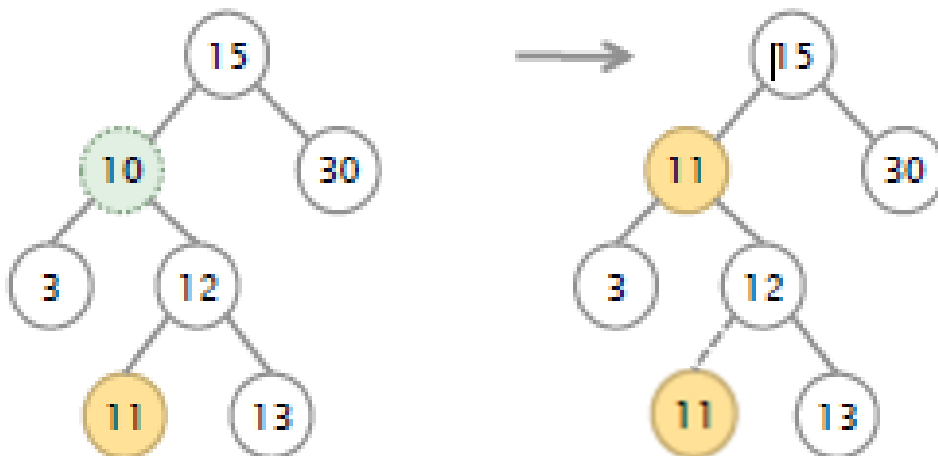
- Excluir um elemento

➤ Caso 3: O nó a ser excluído tem 2 filhos

— Exemplo 2:

Passo 1: Encontrou o nó 11

Passo 2: Copiar valor



Árvores Binárias de Busca

- Excluir um elemento

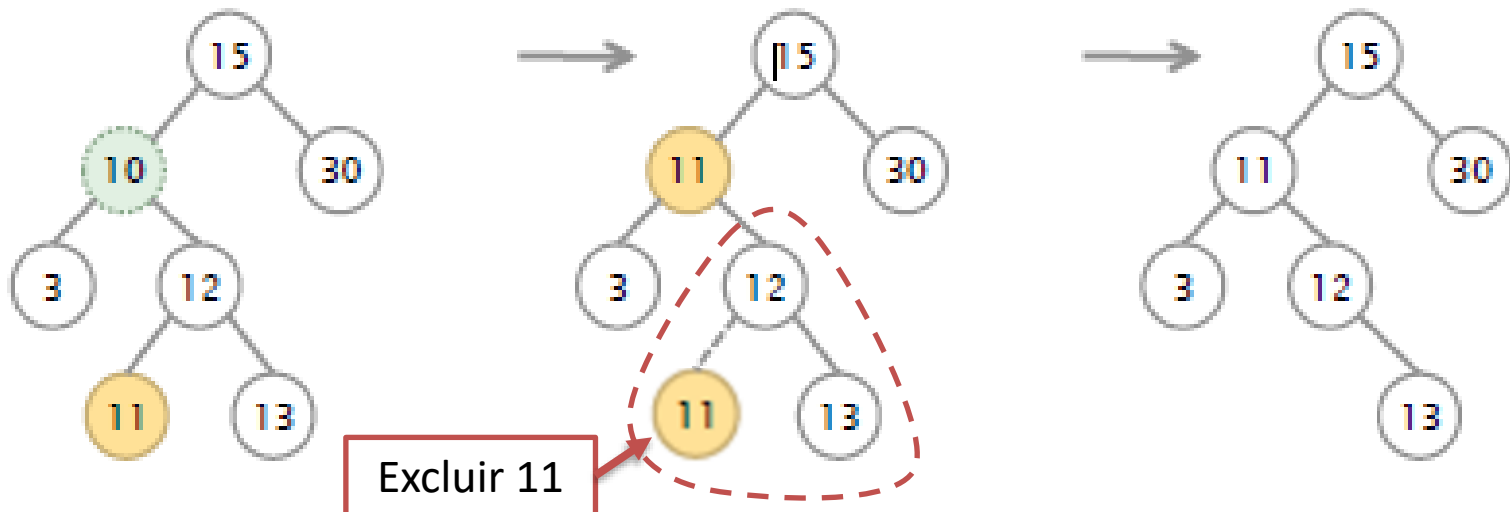
- Caso 3: O nó a ser excluído tem 2 filhos

- Exemplo 2:

Passo 1: Encontrou o nó 11

Passo 2: Copiar valor

Passo 3: Excluir nó 11 da subárvore direita



Árvores Binárias de Busca

- **Excluir um elemento**
 - Algumas observações:
 - O **nó excluído** sempre é **substituído** pelo seu **sucessor**.
 - O **sucessor** é sempre o **nó de menor chave** da **subárvore direita**.

Árvores Binárias de Busca

- **Operações auxiliares**

- Para realizar as **operações básicas**, algumas **outras funções** podem precisar ser criadas.
- Exemplos de **operações auxiliares**:
 - **Mínimo**: Encontrar o menor nó de uma ABB.
 - **Máximo**: Encontrar o maior nó de uma ABB.
 - **Intervalo**: Encontrar todos os elementos de uma árvore dentro de um intervalo.
- No caso da exclusão, por exemplo, é necessário aplicar a operação de **mínimo** na subárvore direita.

Árvores Binárias de Busca

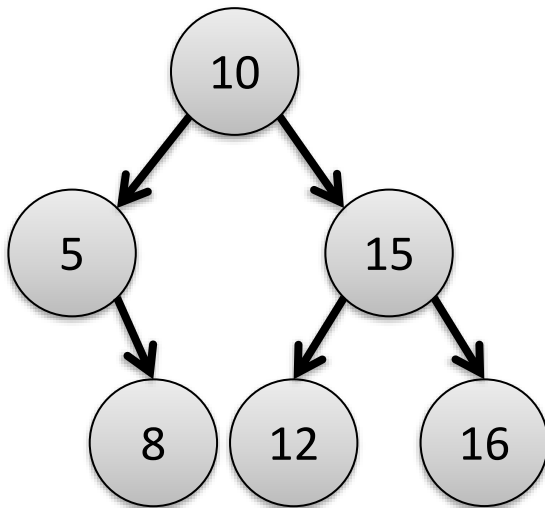
- **Mínimo**
 - Basta **percorrer** todas as **subárvores esquerdas** até que encontre um nó sem filho esquerdo.
 - **Máximo**
 - Basta percorrer todas as **subárvores direitas** até que encontre um nó sem filho direito.
- Ambas as funções possuem funcionamento naturalmente **recursivo**, mas também podem ser implementadas de forma **iterativa**.

Árvores Binárias de Busca

- **Intervalo**
 - Dado um intervalo $[k1, k2]$, deseja-se saber **todos** os elementos da árvore que estão **dentro** desse **intervalo**.
 - Algoritmo:
 1. Começa pela **raiz**.
 2. Se o nó é NULL, **retorna**.
 3. Se o valor do nó **está no intervalo**, aplica a função em seus filhos e imprime o valor.
 4. Se o valor do nó é **menor que k1**, aplica a função no filho da direita.
 5. Se o valor do nó é **maior que k2**, aplica a função no filho da esquerda.

Árvores Binárias de Busca

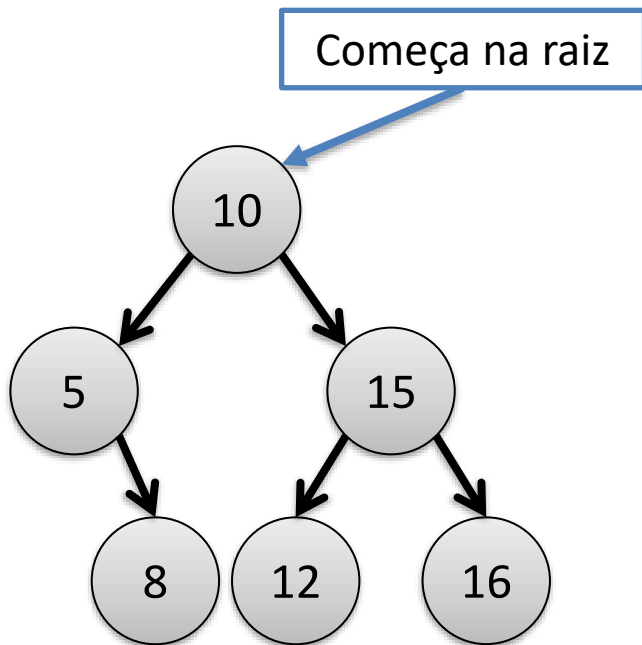
- **Intervalo**
 - Exemplo: Buscar elementos entre 5 e 13.



Árvores Binárias de Busca

- **Intervalo**

- Exemplo: Buscar elementos entre 5 e 13.

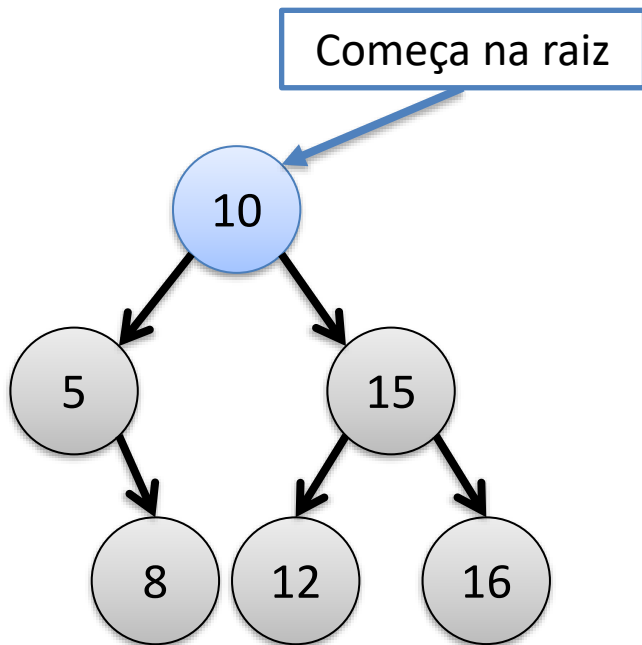


Aplicando em 10:
Valor do nó está no intervalo?

Árvores Binárias de Busca

- **Intervalo**

- Exemplo: Buscar elementos entre 5 e 13.

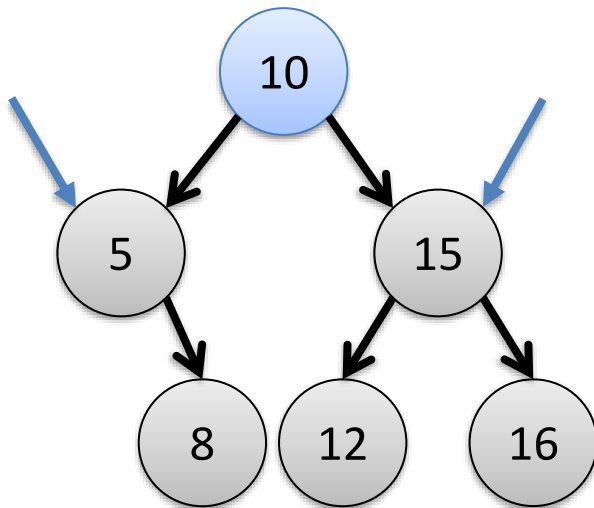


Aplicando em 10:
Valor do nó está no intervalo? **Sim**
Aplica em ambos os nós filhos.

Árvores Binárias de Busca

- **Intervalo**

- Exemplo: Buscar elementos entre 5 e 13.



Aplicando em 10:
Valor do nó está no intervalo? **Sim**
Aplica em ambos os nós filhos.

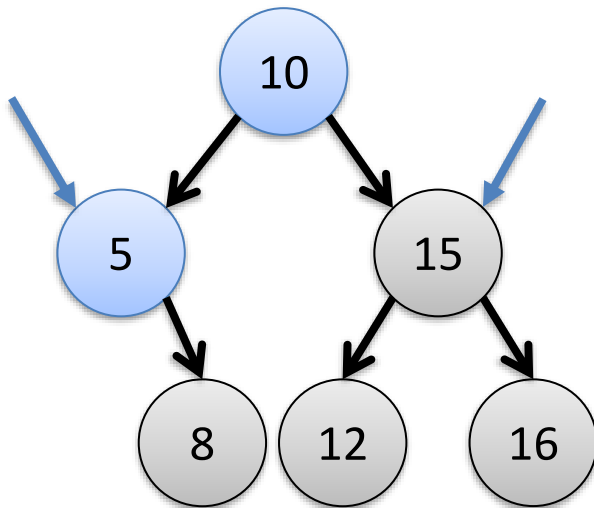
Aplicando em 5:
Valor do nó está no intervalo?

Aplicando em 15:
Valor do nó está no intervalo?

Árvores Binárias de Busca

- **Intervalo**

- Exemplo: Buscar elementos entre 5 e 13.



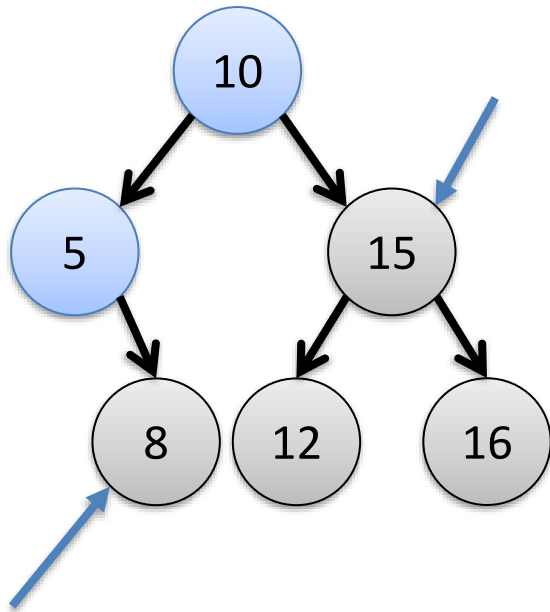
Aplicando em 10:
Valor do nó está no intervalo? **Sim**
Aplica em ambos os nós filhos.

Aplicando em 5:
Valor do nó está no intervalo? **Sim**
Aplica em ambos os nós filhos.

Aplicando em 15:
Valor do nó está no intervalo?

- Intervalo

- Exemplo: Buscar elementos entre 5 e 13.



Aplicando em 10:

Valor do nó está no intervalo? Sim

Aplica em ambos os nós filhos.

Aplicando em 5:

Valor do nó está no intervalo? Sim

Aplica em ambos os nós filhos.

Aplicando em 15:

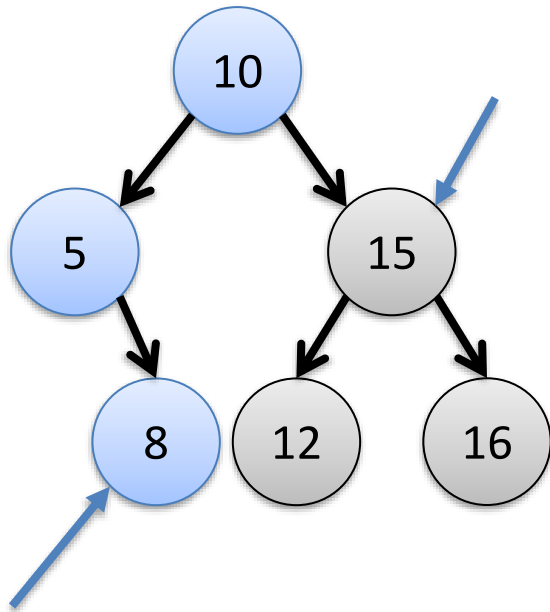
Valor do nó está no intervalo?

Aplicando em 8:

Valor do nó está no intervalo?

- Intervalo

- Exemplo: Buscar elementos entre 5 e 13.



Aplicando em 10:

Valor do nó está no intervalo? Sim

Aplica em ambos os nós filhos.

Aplicando em 5:

Valor do nó está no intervalo? Sim

Aplica em ambos os nós filhos.

Aplicando em 15:

Valor do nó está no intervalo?

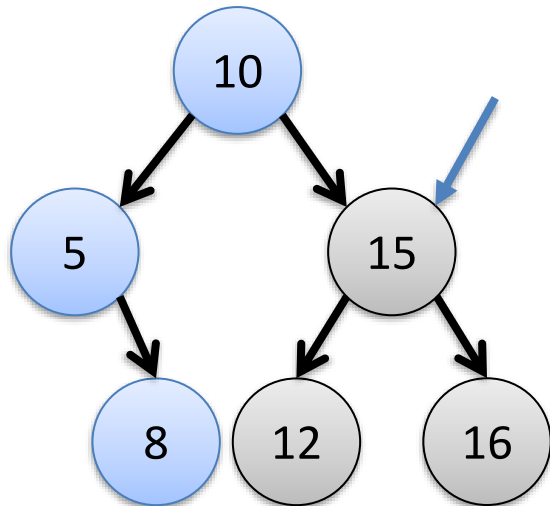
Aplicando em 8:

Valor do nó está no intervalo? Sim

Aplica em ambos os nós filhos.

- Intervalo

- Exemplo: Buscar elementos entre 5 e 13.



8,

Aplicando em 10:

Valor do nó está no intervalo? Sim

Aplica em ambos os nós filhos.

Aplicando em 5:

Valor do nó está no intervalo? Sim

Aplica em ambos os nós filhos.

Aplicando em 15:

Valor do nó está no intervalo?

Aplicando em 8:

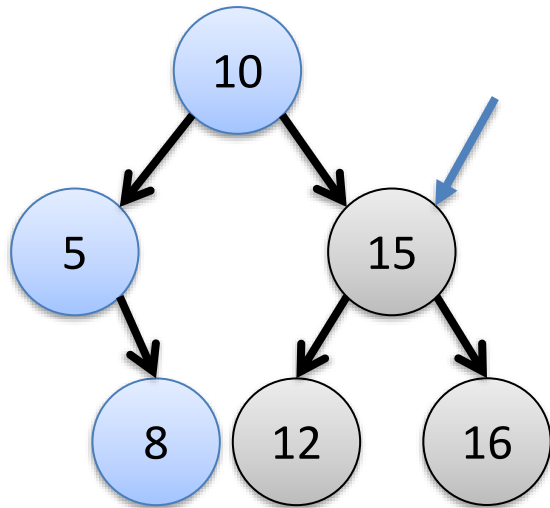
Valor do nó está no intervalo? Sim

Aplica em ambos os nós filhos.

Imprime

- Intervalo

- Exemplo: Buscar elementos entre 5 e 13.



8, 5,

Aplicando em 10:
Valor do nó está no intervalo? **Sim**
Aplica em ambos os nós filhos.

Aplicando em 5:
Valor do nó está no intervalo? **Sim**
Aplica em ambos os nós filhos.

→ Imprime

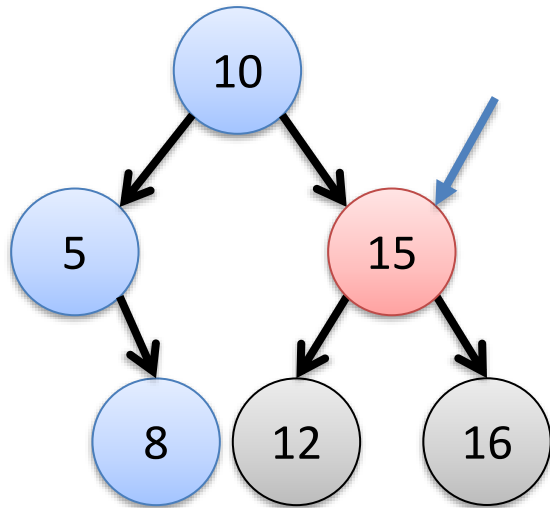
Aplicando em 15:
Valor do nó está no intervalo?

Aplicando em 8:
Valor do nó está no intervalo? **Sim**
Aplica em ambos os nós filhos.

→ Imprime

- Intervalo

- Exemplo: Buscar elementos entre 5 e 13.



8, 5,

Aplicando em 10:

Valor do nó está no intervalo? Sim

Aplica em ambos os nós filhos.

Aplicando em 5:

Valor do nó está no intervalo? Sim

Aplica em ambos os nós filhos.

Imprime

Aplicando em 15:

Valor do nó está no intervalo? Não

15 > 13, logo, aplicar no filho esq.

Aplicando em 8:

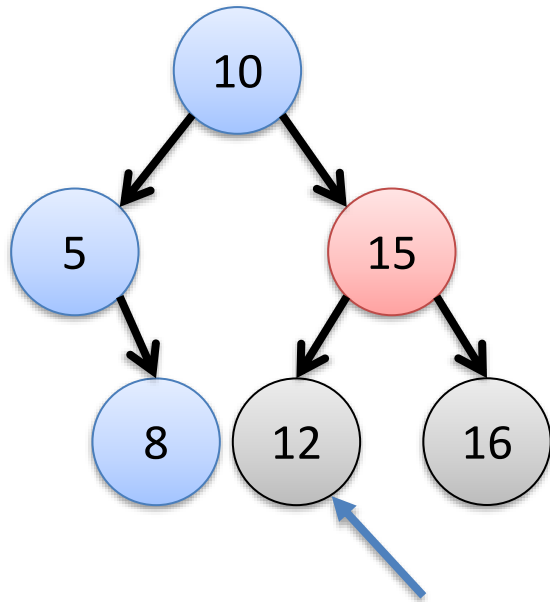
Valor do nó está no intervalo? Sim

Aplica em ambos os nós filhos.

Imprime

- Intervalo

- Exemplo: Buscar elementos entre 5 e 13.



8, 5,

Aplicando em 10:
Valor do nó está no intervalo? **Sim**
Aplica em ambos os nós filhos.

Aplicando em 5:
Valor do nó está no intervalo? **Sim**
Aplica em ambos os nós filhos.

→ Imprime

Aplicando em 15:
Valor do nó está no intervalo? **Não**
15 > 13, logo, aplicar no filho esq.

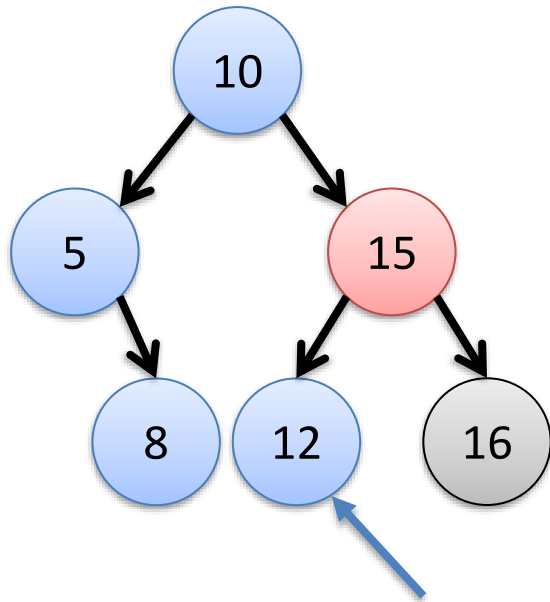
Aplicando em 8:
Valor do nó está no intervalo? **Sim**
Aplica em ambos os nós filhos.

→ Imprime

Aplicando em 12:
Valor do nó está no intervalo?

- Intervalo

- Exemplo: Buscar elementos entre 5 e 13.



8, 5,

Aplicando em 10:

Valor do nó está no intervalo? Sim

Aplica em ambos os nós filhos.

Aplicando em 5:

Valor do nó está no intervalo? Sim

Aplica em ambos os nós filhos.

Imprime

Aplicando em 15:

Valor do nó está no intervalo? Não

15>13, logo, aplicar no filho esq.

Aplicando em 8:

Valor do nó está no intervalo? Sim

Aplica em ambos os nós filhos.

Imprime

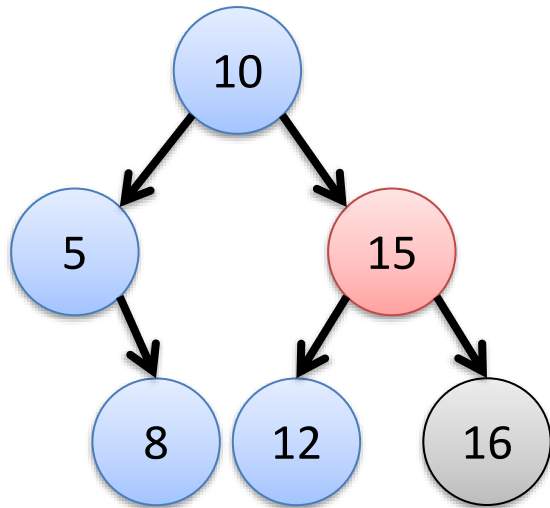
Aplicando em 12:

Valor do nó está no intervalo? Sim

Aplica em ambos os nós filhos.

- Intervalo

- Exemplo: Buscar elementos entre 5 e 13.



8, 5, 12,

Aplicando em 10:
Valor do nó está no intervalo? **Sim**
Aplica em ambos os nós filhos.

Aplicando em 5:
Valor do nó está no intervalo? **Sim**
Aplica em ambos os nós filhos.

Imprime

Aplicando em 15:
Valor do nó está no intervalo? **Não**
15 > 13, logo, aplicar no filho esq.

Aplicando em 8:
Valor do nó está no intervalo? **Sim**
Aplica em ambos os nós filhos.

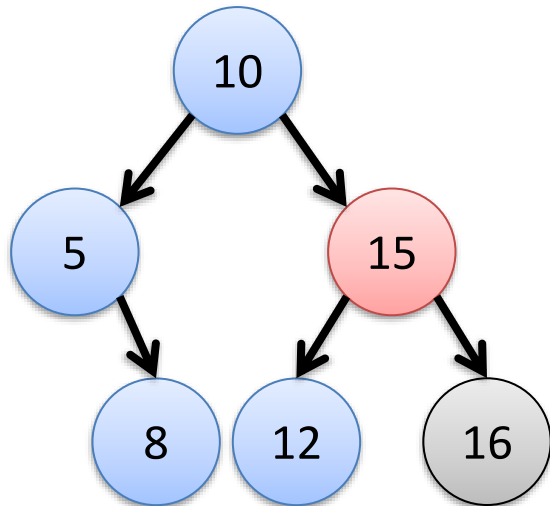
Imprime

Aplicando em 12:
Valor do nó está no intervalo? **Sim**
Aplica em ambos os nós filhos.

Imprime

- Intervalo

- Exemplo: Buscar elementos entre 5 e 13.



8, 5, 12,

Aplicando em 10:
Valor do nó está no intervalo? **Sim**
Aplica em ambos os nós filhos.

Aplicando em 5:
Valor do nó está no intervalo? **Sim**
Aplica em ambos os nós filhos.

Imprime

Aplicando em 15:
Valor do nó está no intervalo? **Não**
15 > 13, logo, aplicar no filho esq.

Retorna

Aplicando em 8:
Valor do nó está no intervalo? **Sim**
Aplica em ambos os nós filhos.

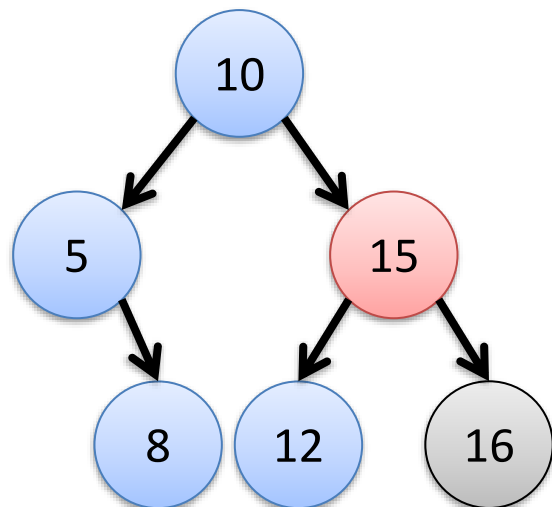
Imprime

Aplicando em 12:
Valor do nó está no intervalo? **Sim**
Aplica em ambos os nós filhos.

Imprime

- Intervalo

- Exemplo: Buscar elementos entre 5 e 13.



8, 5, 12, 10

Aplicando em 10:

Valor do nó está no intervalo? Sim

Aplica em ambos os nós filhos.

Imprime

Aplicando em 5:

Valor do nó está no intervalo? Sim

Aplica em ambos os nós filhos.

Imprime

Aplicando em 15:

Valor do nó está no intervalo? Não

15 > 13, logo, aplicar no filho esq.

Retorna

Aplicando em 8:

Valor do nó está no intervalo? Sim

Aplica em ambos os nós filhos.

Imprime

Aplicando em 12:

Valor do nó está no intervalo? Sim

Aplica em ambos os nós filhos.

Imprime

Árvores Binárias de Busca

- **Exercício 2:**

- a) Converta o seguinte vetor em uma ABB, seguindo a ordem mostrada:

15	5	16	3	12	20	10	6	13	7	23	18
----	---	----	---	----	----	----	---	----	---	----	----

- b) Remova, na ordem mostrada, os seguintes elementos:

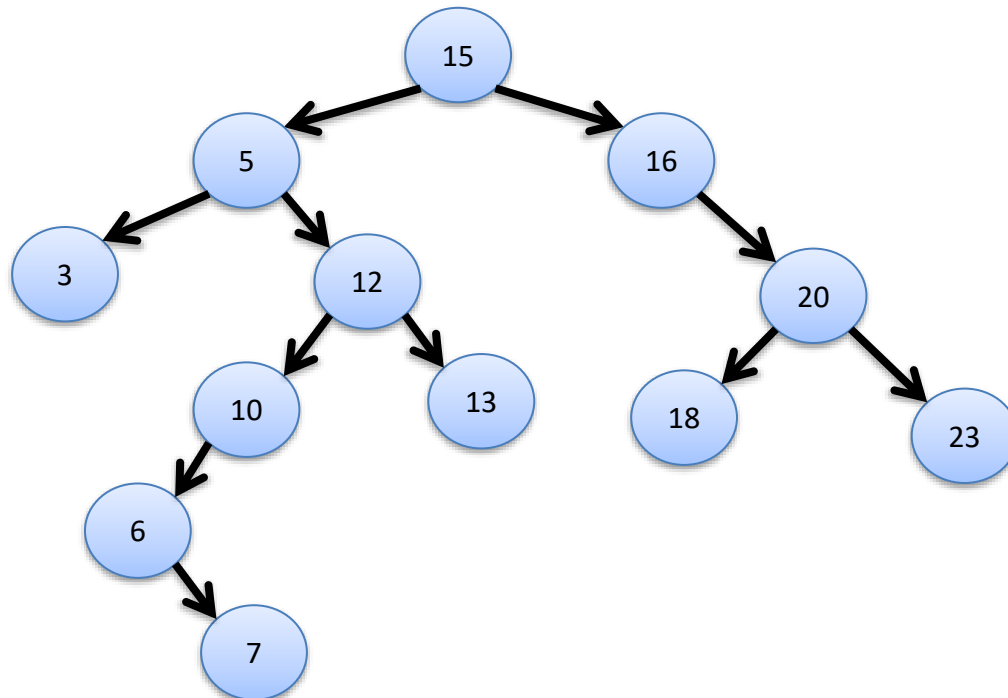
5, 20, 12, 15

- c) Depois, aplique o algoritmo para determinar os elementos da ABB que estão no intervalo [1, 10]
- d) Por fim, imprima todos os elementos da árvore resultante em pré-ordem.

- **Exercício 2:**

a) Converta o seguinte vetor em uma ABB, seguindo a ordem mostrada:

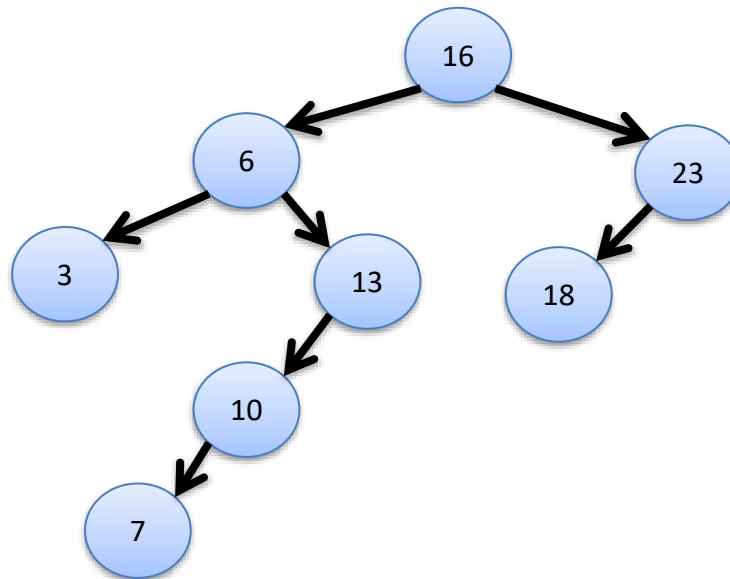
15	5	16	3	12	20	10	6	13	7	23	18
----	---	----	---	----	----	----	---	----	---	----	----



- **Exercício 2:**

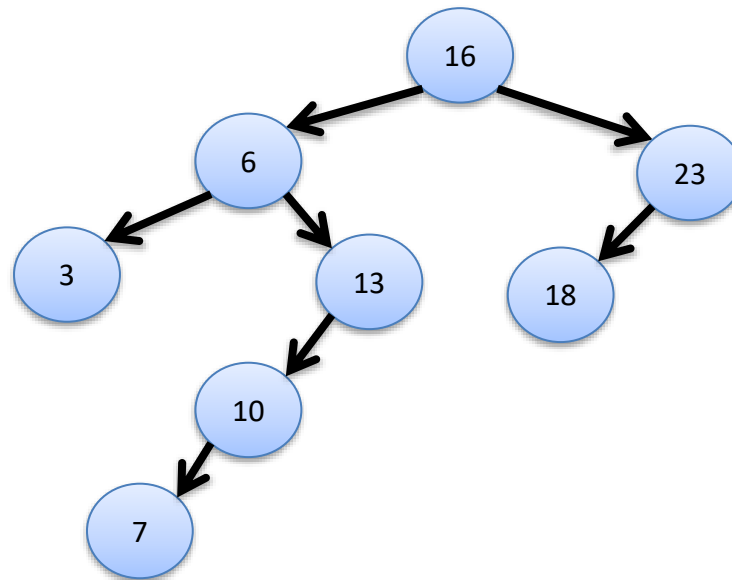
b) Remova, na ordem mostrada, os seguintes elementos:

5, 20, 12, 15



- **Exercício 2:**

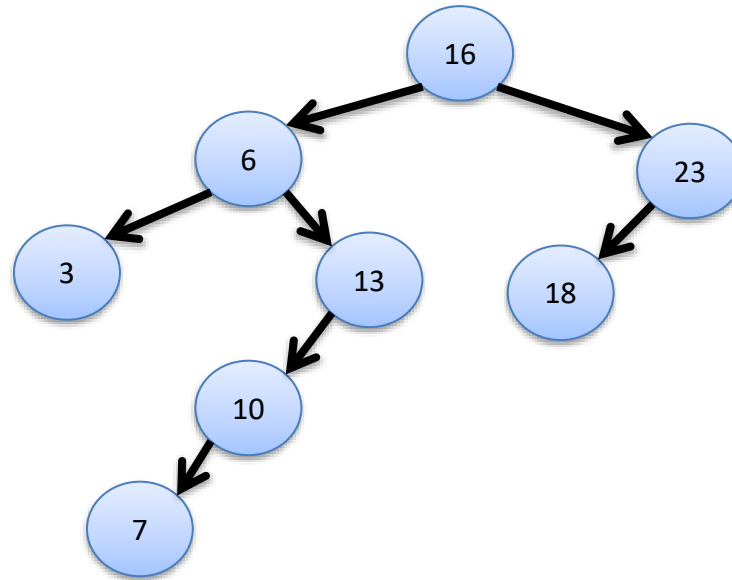
- c) Depois, aplique o algoritmo para determinar os elementos da ABB que estão no intervalo $[1, 10]$



3, 7, 10, 6

- **Exercício 2:**

d) Por fim, imprima todos os elementos da árvore resultante em pré-ordem.



16, 6, 3, 13, 10, 7, 23, 18

Árvores Binárias de Busca

FIM