



ÁRVORE BINÁRIA DE BUSCA

Prof. Cristhiane

ÁRVORE

- Conjunto finito de N estruturas elementares chamadas nós, tal que:
 - Existe um nó especial que é o **nó raiz**
 - Os restantes $N-1$ nós são divididos em M conjuntos disjuntos (S_1, S_2, \dots, S_M) , onde cada um desses conjuntos é também uma árvore.
- Assim como as listas lineares, árvores são estruturas de dados que caracterizam uma relação entre os dados que a compõem.
- No caso de árvores, a relação existente entre os nós é uma relação de hierarquia, onde um conjunto de **nós é hierarquicamente subordinado** a outro.

REPRESENTAÇÕES GRÁFICAS

Grafo

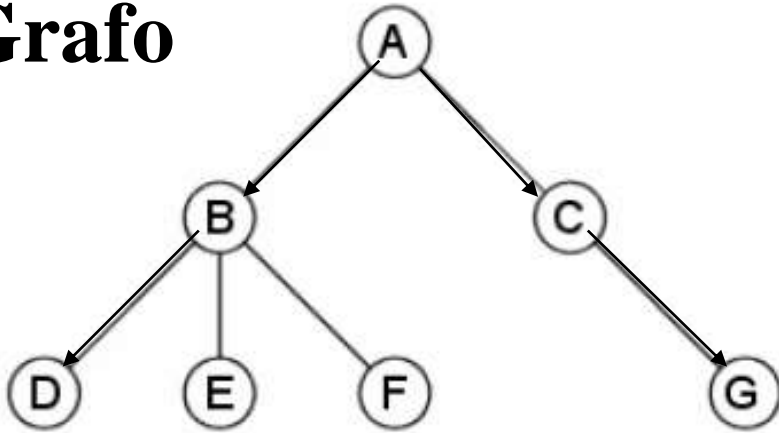
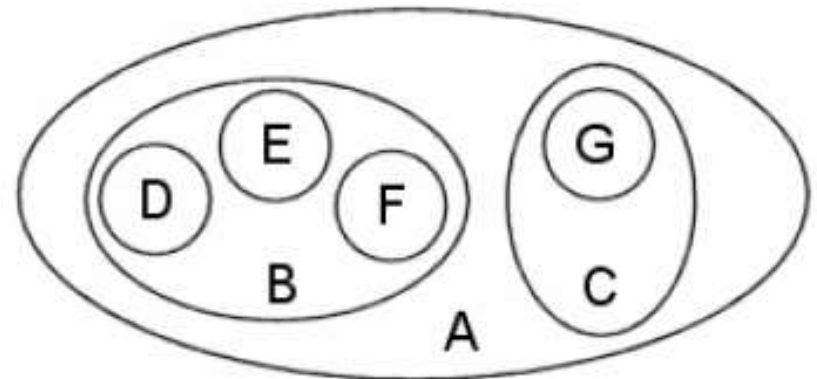


Diagrama de Venn

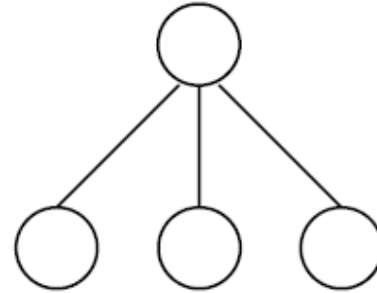


EXEMPLOS DE ÁRVORES

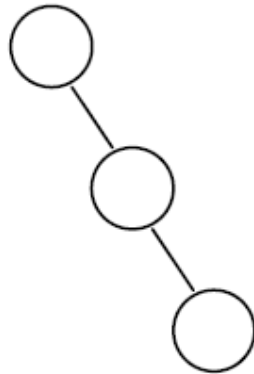
a)



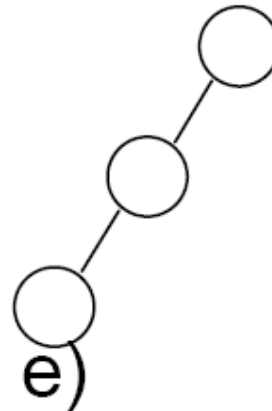
b)



c)



d)



e)



TERMINOLOGIA

- A **raiz** de uma árvore é chamada de **pai** de suas subárvores.
- Nós com o mesmo pai são denominados **irmãos**.
- O **grau** de um nó é por definição o número de subárvores do nó.
- O **grau da árvore** é o grau máximo entre todos os nós.
- Um **nó folha** ou **terminal** tem grau zero, ou seja, não tem sub-árvores.



TERMINOLOGIA

- O nó raiz tem **nível** um, os nós filhos da raiz têm nível 2 e, assim, sucessivamente.
- A **altura** ou **profundidade** de uma árvore é igual ao número de níveis desta árvore.
- Os nós que não são raiz ou folhas, são chamados **nós internos**.
- Um conjunto de árvores disjuntas forma uma **floresta**.



ÁRVORE BINÁRIA

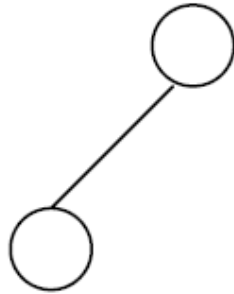
- Cada nó da árvore pode ter no máximo duas subárvores, ou seja, **o grau de cada nó pode ser 0, 1 ou 2.**
- Com grau 2, cada nó (com exceção das folhas) terá uma subárvore **esquerda (E)** e uma subárvore **direita (D)**.



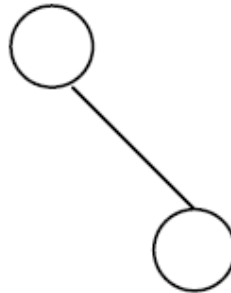
EXEMPLOS DE ÁRVORES BINÁRIAS



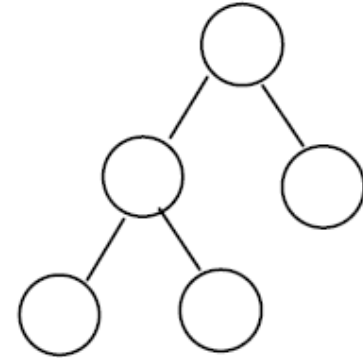
a)



b)



c)

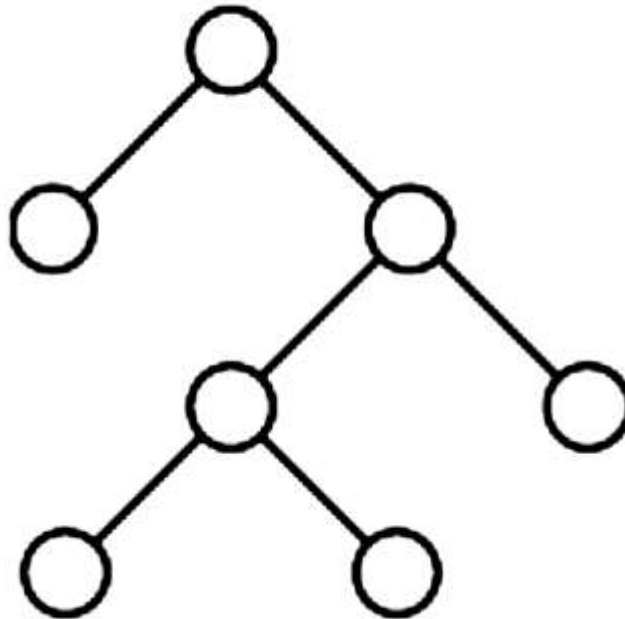


d)



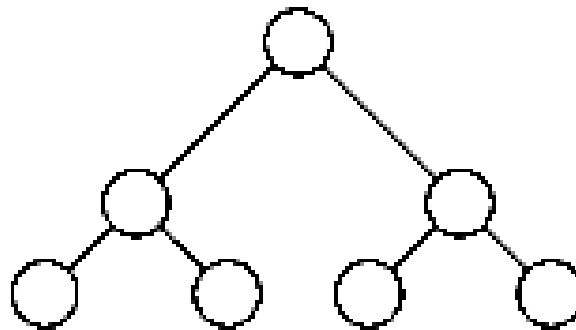
ÁRVORE ESTRITAMENTE BINÁRIA

- Cada nó tem 0 ou 2 sub-árvores, ou seja, nenhum nó tem filho único.



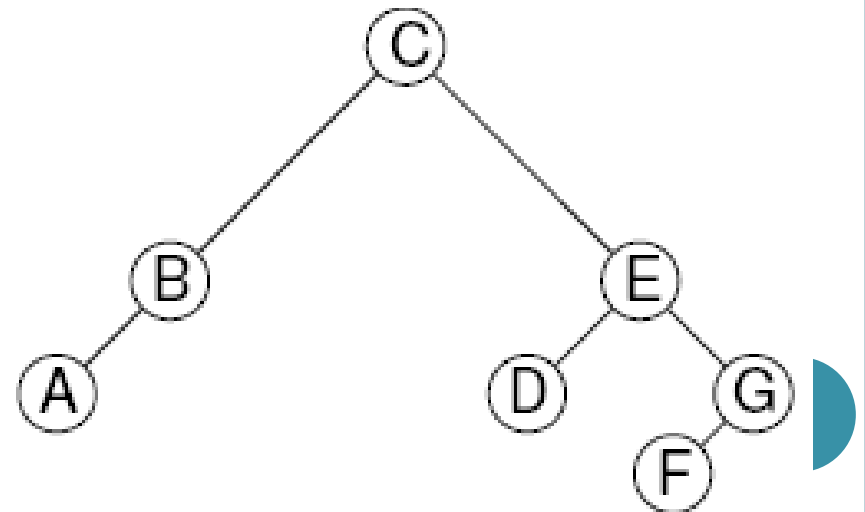
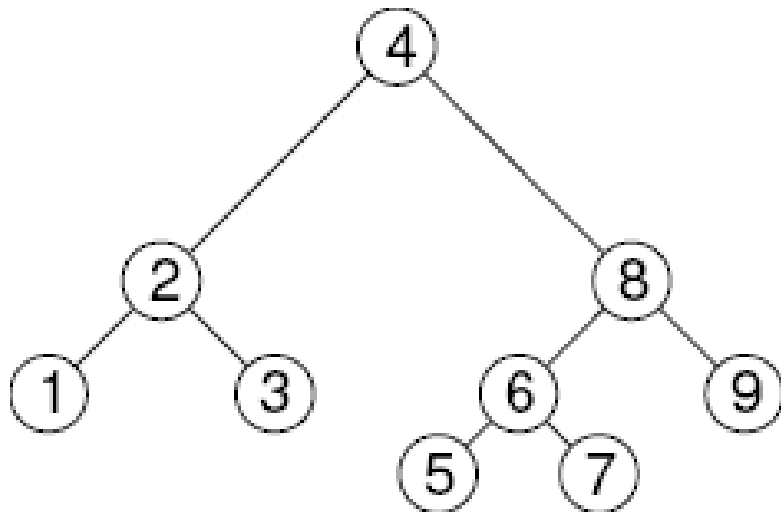
ÁRVORE BINÁRIA COMPLETA

- Uma árvore completa é uma árvore estritamente binária (grau 0 ou 2), na qual todos os nós folhas estão no último nível.
- O número de nós de uma árvore binária completa é $2^h - 1$, em que h é a sua altura.



ÁRVORE BINÁRIA DE BUSCA (PESQUISA)

- Árvore binária em que **todo nó tem chave maior que a chave dos seus descendentes à esquerda e menor que a chave dos seus descendentes à direita.**

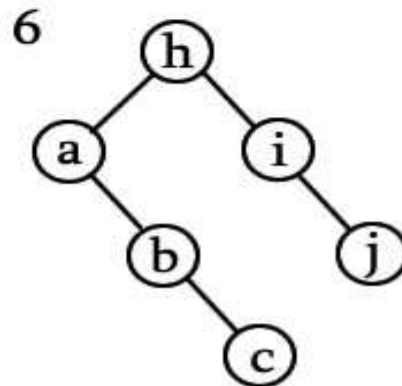
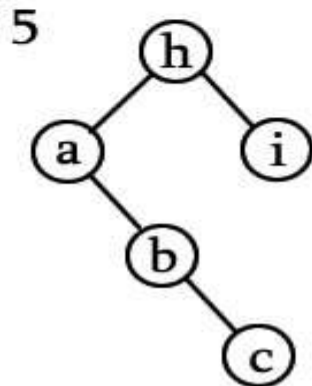
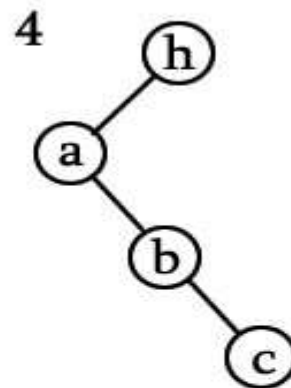
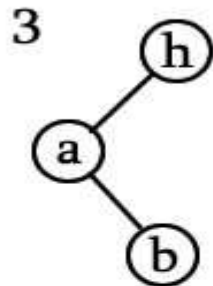
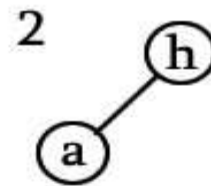
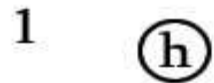


PESQUISA EM UMA ÁRVORE BINÁRIA DE BUSCA

- Para encontrar um registro com uma chave x , compare-a com a chave que está na raiz.
 - Se for menor, vá para a subárvore esquerda;
 - Se for maior, vá para a subárvore direita.
- Repita o processo recursivamente até que a chave procurada seja encontrada ou um nó folha seja atingido.



INSERÇÃO EM ÁRVORE BINÁRIA DE BUSCA



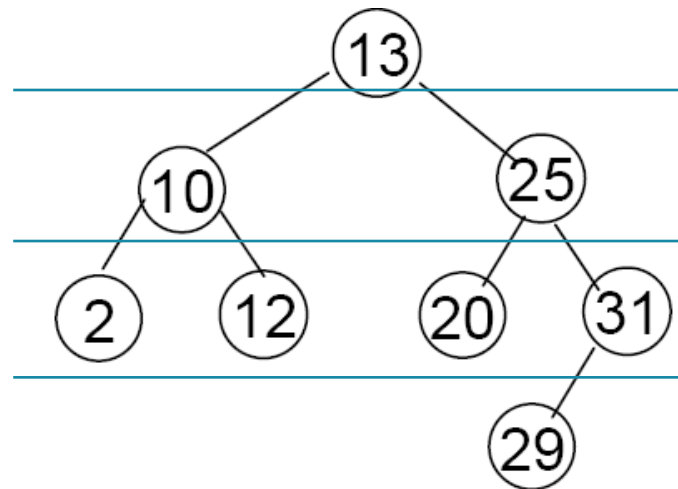
PERCURSOS EM ÁRVORES

- Percurso é o processo de visitar cada nó somente uma vez.
- Pode ser interpretado como colocar todos os nós numa linha ou a linearização de uma árvore.
- Existem basicamente dois tipos de percursos:
 - Percurso em largura (ou extensão)
 - Percurso em profundidade



PERCURSO EM LARGURA

- Consiste em visitar cada nó começando do nível mais alto e movendo para baixo nível a nível, visitando nós em cada nível da esquerda para a direita.
- Exemplo:



- O exemplo da figura acima resulta na sequência 13,10,25,2,12,20,31,29.



PERCURSO EM PROFUNDIDADE

- Prossegue tanto quanto possível à esquerda, então se move para trás até a primeira encruzilhada, vai um passo à direita e novamente tanto quanto possível para a esquerda.

- Variações do percurso em profundidade:

- VED: pré-ordem
- EVD: em-ordem
- EDV: pós-ordem

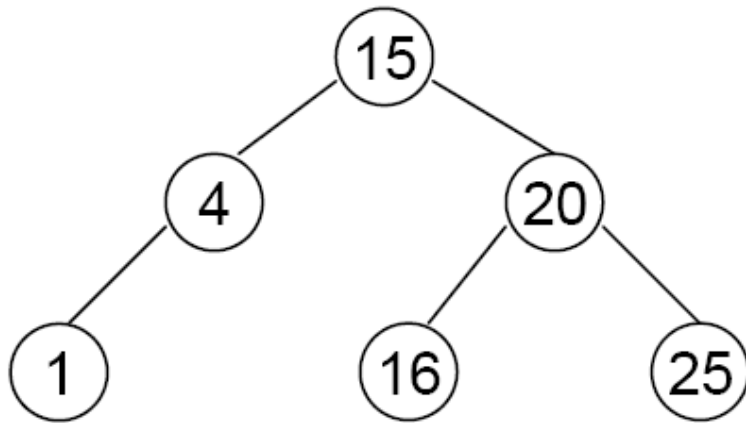
V = visitar um nó

E = percorrer a sub-árvore esquerda (Left)

D = percorrer a sub-árvore direita (Right)



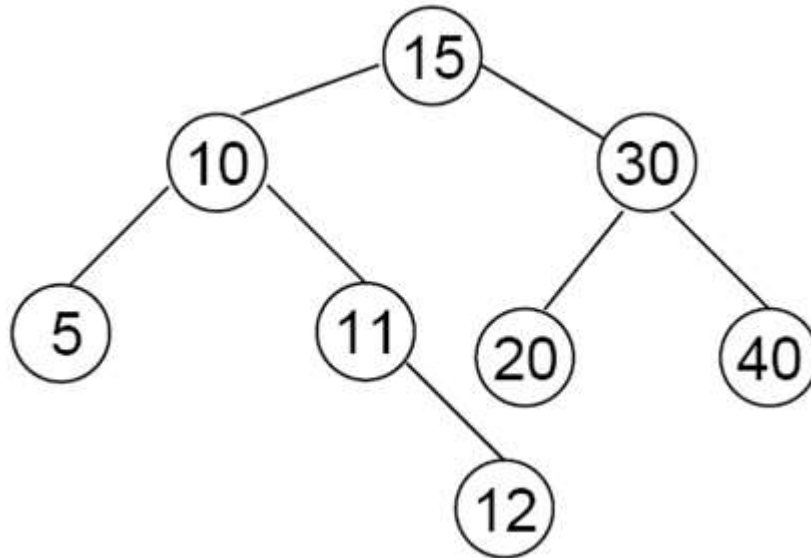
EXEMPLO 1 - PERCURSO PROFUNDIDADE



- Pré-ordem: 15, 4, 1, 20, 16, 25
- Em-ordem: 1, 4, 15, 16, 20, 25
- Pós-ordem: 1, 4, 16, 25, 20, 15



EXEMPLO 2 - PERCURSO PROFUNDIDADE



- Pré-ordem: 15, 10, 5, 11, 12, 30, 20, 40
- Em-ordem: 5, 10, 11, 12, 15, 20, 30, 40
- Pós-ordem: 5, 12, 11, 10, 20, 40, 30, 15



REMOÇÃO DE UM ELEMENTO

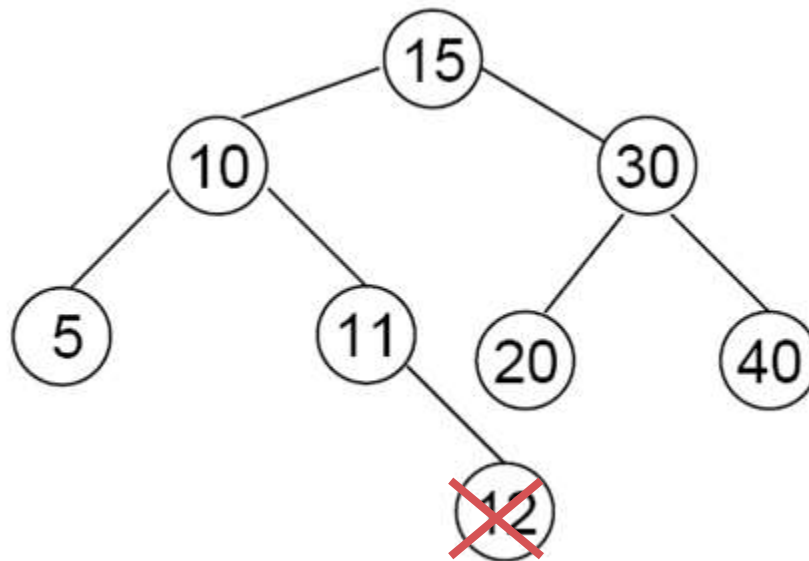
- A remoção de um registro não é tão simples quanto a inserção.
- Assim como a inserção, a remoção só pode ser realizada nas folhas.
- A remoção possui basicamente três casos:
 1. O nó a ser removido é uma folha
 2. O nó a ser removido tem apenas 1 filho
 3. O nó a ser removido é um nó interno com dois filhos/subárvores



REMOÇÃO DE UM ELEMENTO

1. Nó-folha: remoção direta

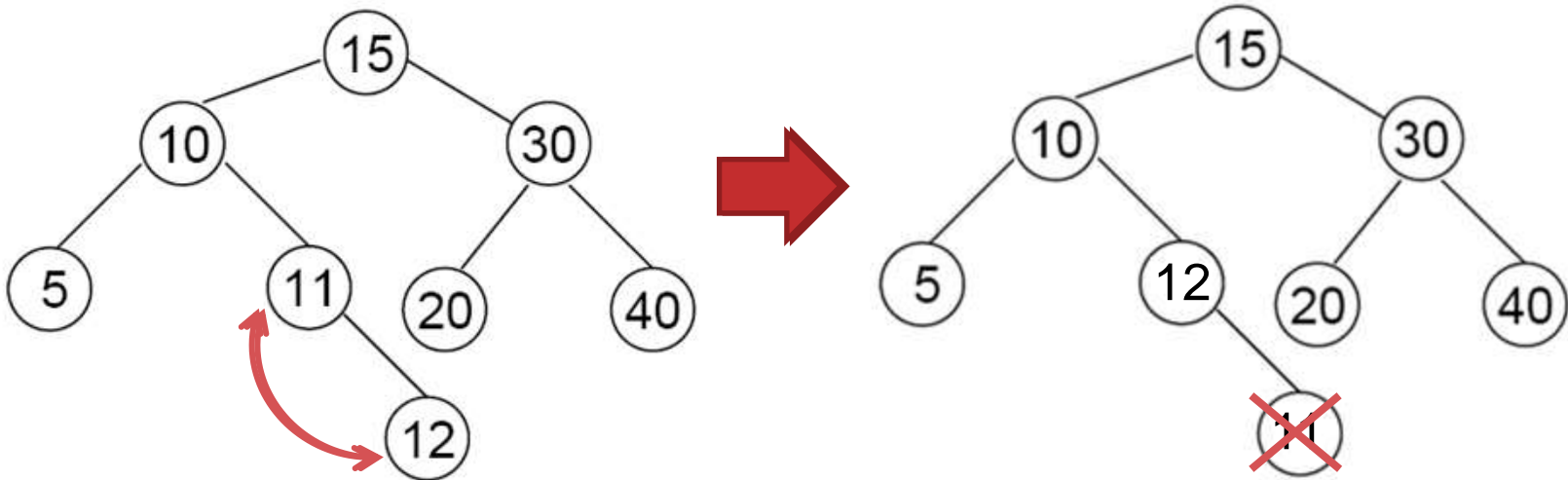
- Exemplo: removendo o nó que contém o número 12



REMOÇÃO DE UM ELEMENTO

2. Nó com apenas 1 filho:

- Se o nó que contém o registro a ser retirado possui no máximo um descendente, a operação é simples, troca-se o nó pai pelo filho e remove.
- Exemplo: remover o nó 11

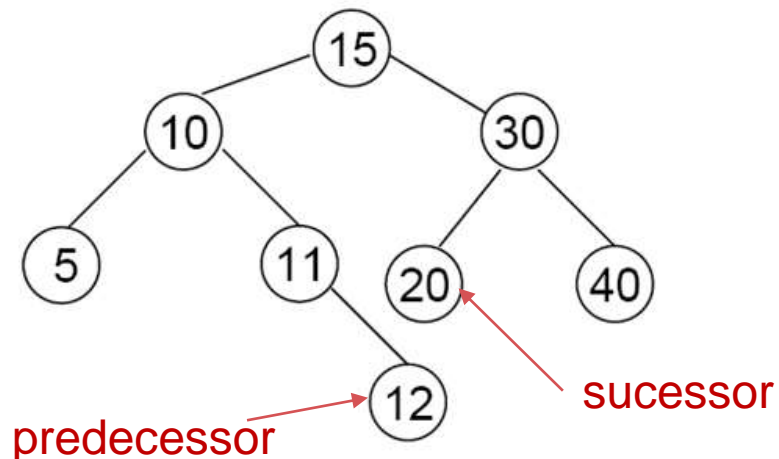


REMOÇÃO DE UM ELEMENTO

3. Nó com dois filhos/subárvores:

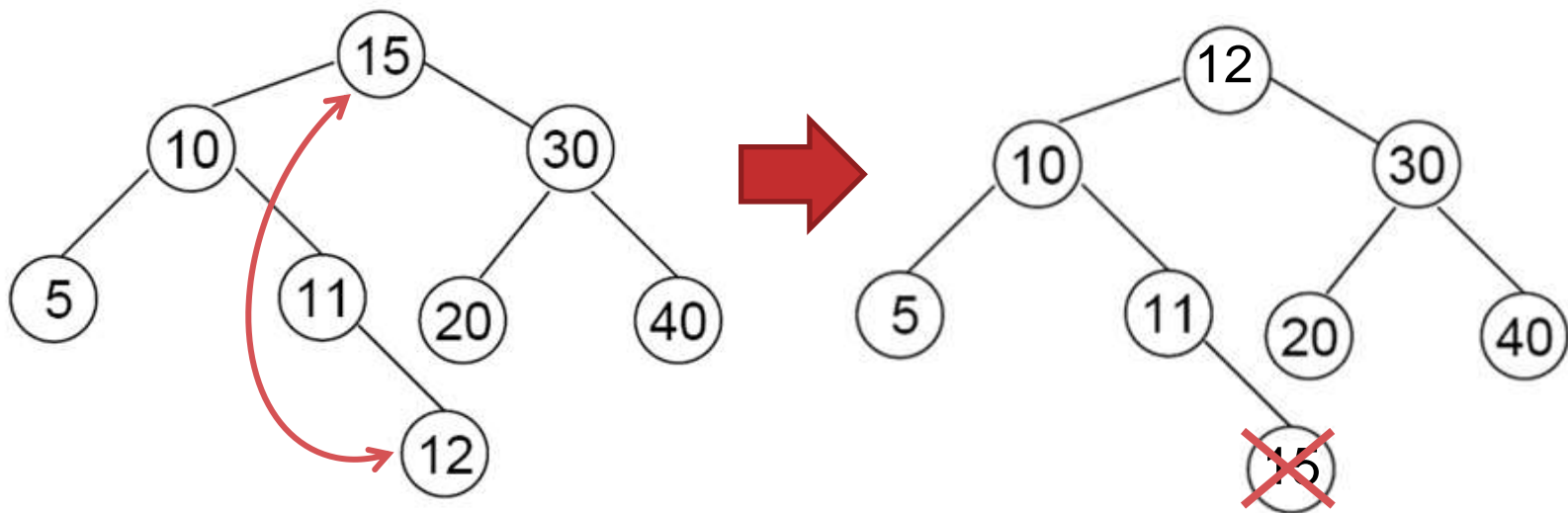
- No caso do nó conter dois descendentes o registro a ser retirado deve ser primeiro:
 - substituído pelo registro mais à direita da subárvore esquerda (antecessor);
 - ou pelo registro mais à esquerda na subárvore direita (predecessor).

Exemplo: remover o nó 15



REMOÇÃO DE UM ELEMENTO

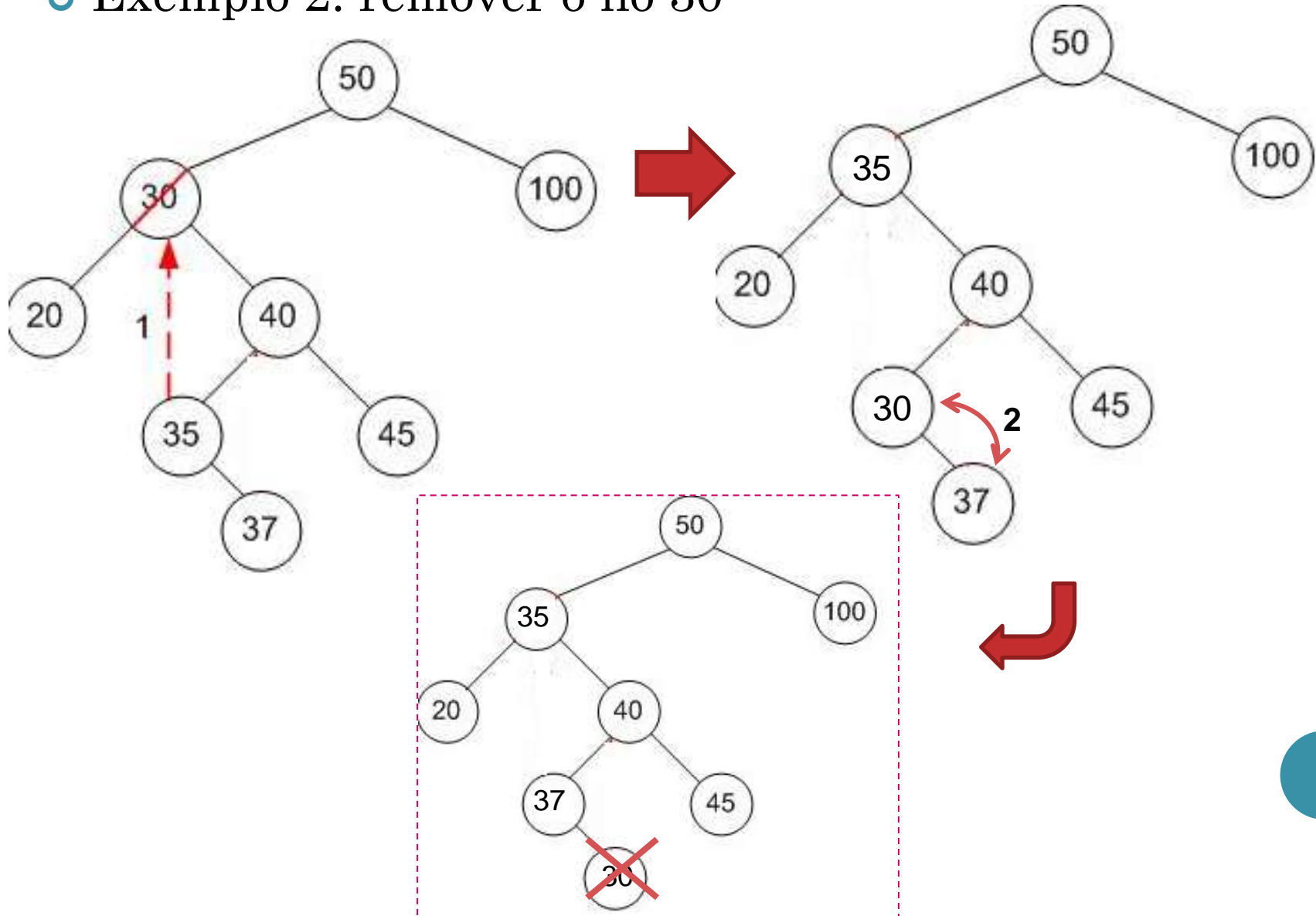
3. Nó com dois filhos/subárvores: removendo o nó 15



- para retirar o registro com chave 15 na árvore basta trocá-lo pelo registro com chave 12 ou pelo registro com chave 20, e então retirar o nó que recebeu o registro com chave 15.

REMOÇÃO DE UM ELEMENTO

- Exemplo 2: remover o nó 30



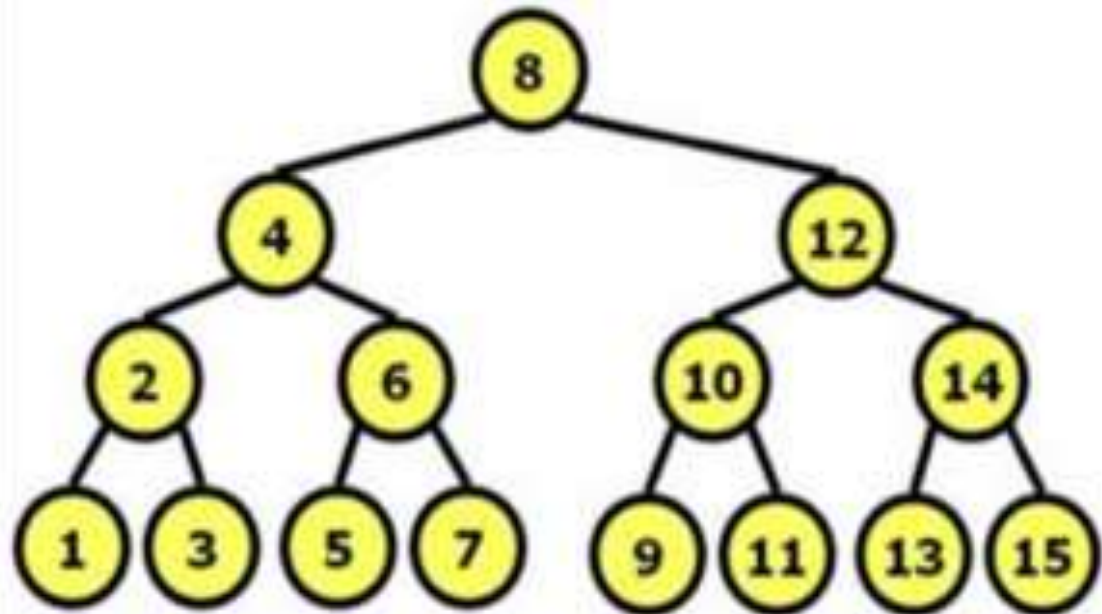
EXERCÍCIOS

- 1) Inserir as chaves 5, 3, 8, 10, 4, 7, 15, 2, 6, 9, 20.
- 2) Inserir as chaves 8, 4, 3, 10, 12, 2, 6, 7, 11, 9.
- 3) Determine a altura da árvore obtida nos exercícios anteriores.



EXERCÍCIOS

- 4) Considere a Árvore Binária de Busca a seguir, faça as remoções dos seguintes nós, na ordem em que eles aparecem, redesenhando a árvore a cada remoção: 8, 3, 10, 6, 12.
- Em seguida, apresente os percursos;
 - pré-ordem
 - em-ordem
 - pós-ordem



LINK PARA A AULA

- <https://drive.google.com/file/d/1ztWjytvTShR7P1dIckgTuSgSMTYIzMfM/view?usp=sharing>

