
Estrutura de Dados

Árvores Binárias

Profa.: Márcia Sampaio Lima

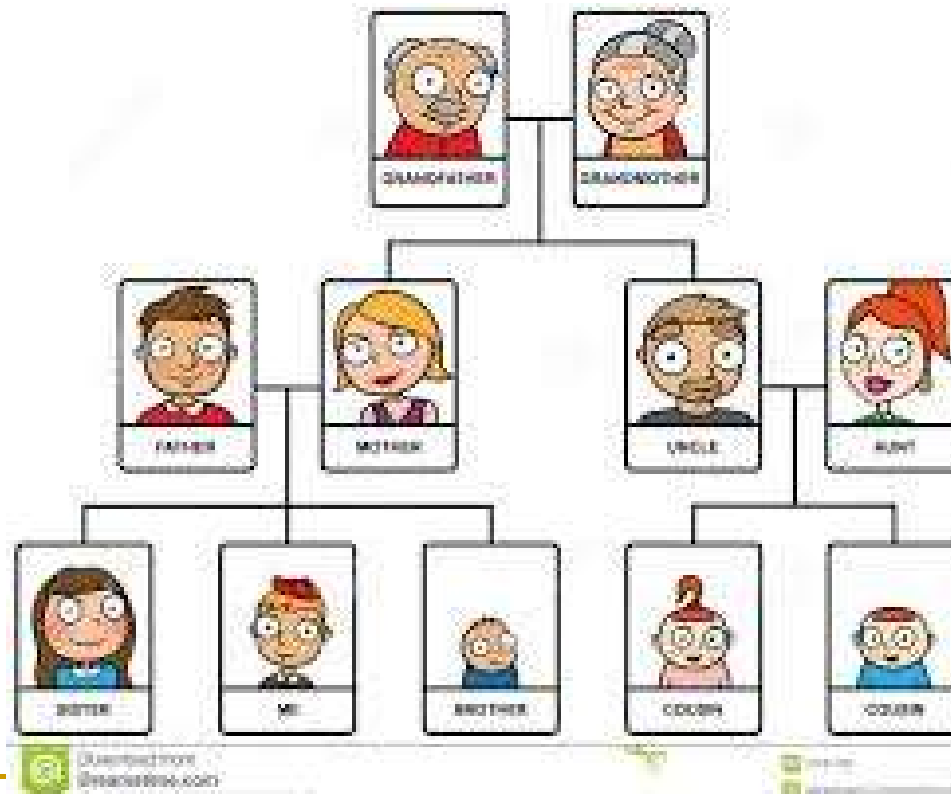
EST - UEA

Árvores

- As estruturas anteriores são chamadas de unidimensionais (ou lineares)
 - Exemplo: vetores e listas
 - Não adequadas para representar hierarquias.
 - Exemplo: árvore de diretórios
 - No cotidiano, diversas informações são organizadas de forma hierárquica. Como exemplo, podem ser citados:
 - o organograma de uma empresa;
 - a divisão de um livro em capítulos, seções, tópicos;
 - a árvore genealógica de uma pessoa.
-

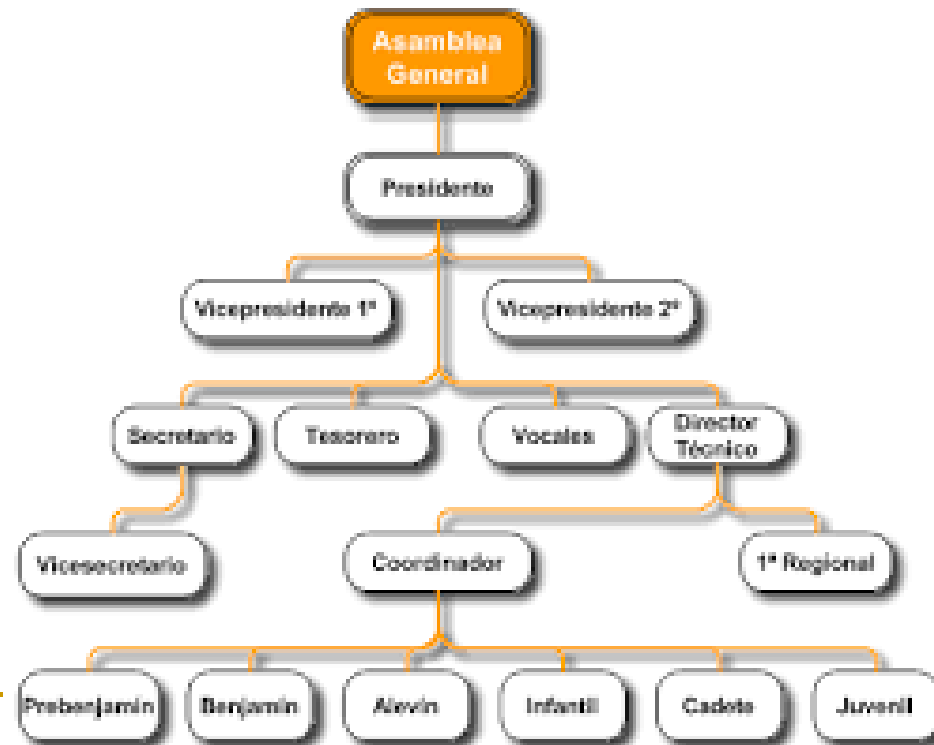
Árvores

■ Árvore Genealógica



Árvores

- Organograma de uma empresa



Árvores

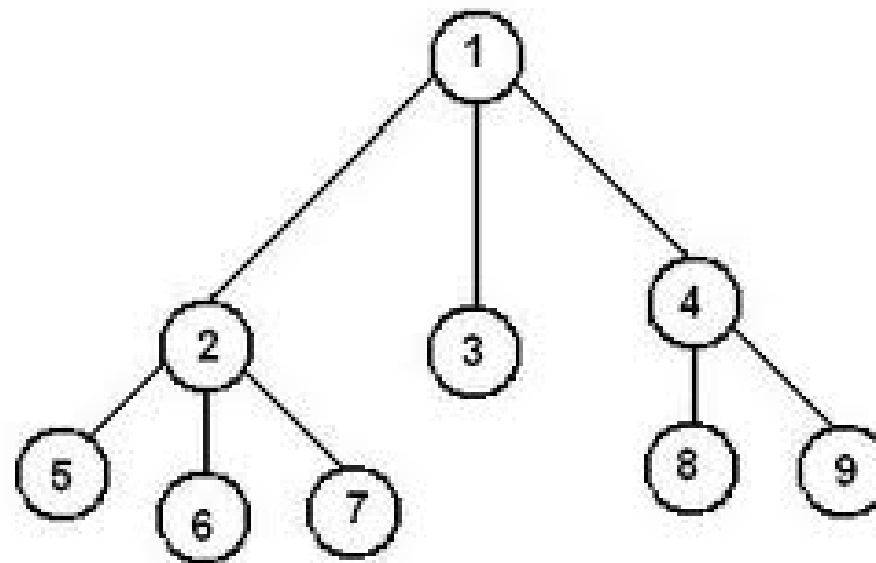
- A principal objeção a listas encadeadas é o acesso seqüencial a seus itens.
 - E se ao invés de um, tivéssemos dois ponteiros em cada item?
 - E se ao invés de começar a inserir no início ou final da lista, resolvêssemos começar a lista pelo centro?
 - Será que eu posso ver uma lista de forma diferente?
 - Árvore é uma estrutura adequada para representar hierarquias.
 - A forma mais natural para definirmos uma estrutura de árvore é usando recursividade.
-

Árvores

■ Uma árvore:

- ❑ Organiza um conjunto de acordo com uma estrutura hierárquica.
 - ❑ Contém elementos que são chamados de nós
 - ❑ O “pai de todos” é a raiz – 1º. da hierarquia
 - ❑ O conteúdo de um nó pode ser de qualquer tipo que se deseje representar.
-

Árvores

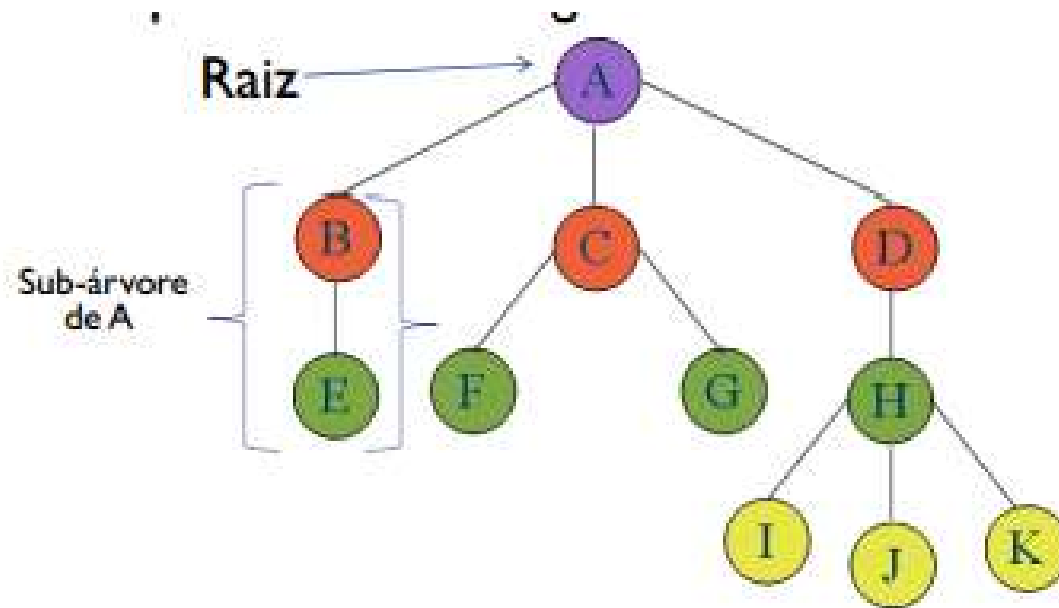


Árvores

■ Estrutura:

- Uma árvore é composta por um conjunto de nós.
 - Existe um nó r , denominado raiz, que contém zero ou mais sub-árvores, cujas raízes são ligadas a r .
 - Esses nós raízes das sub-árvores são ditos filhos do nó pai, no caso r .
 - Nós com filhos são comumente chamados de nós **internos** e nós que não têm filhos são chamados de **folhas**, ou nós externos.
-

Árvores

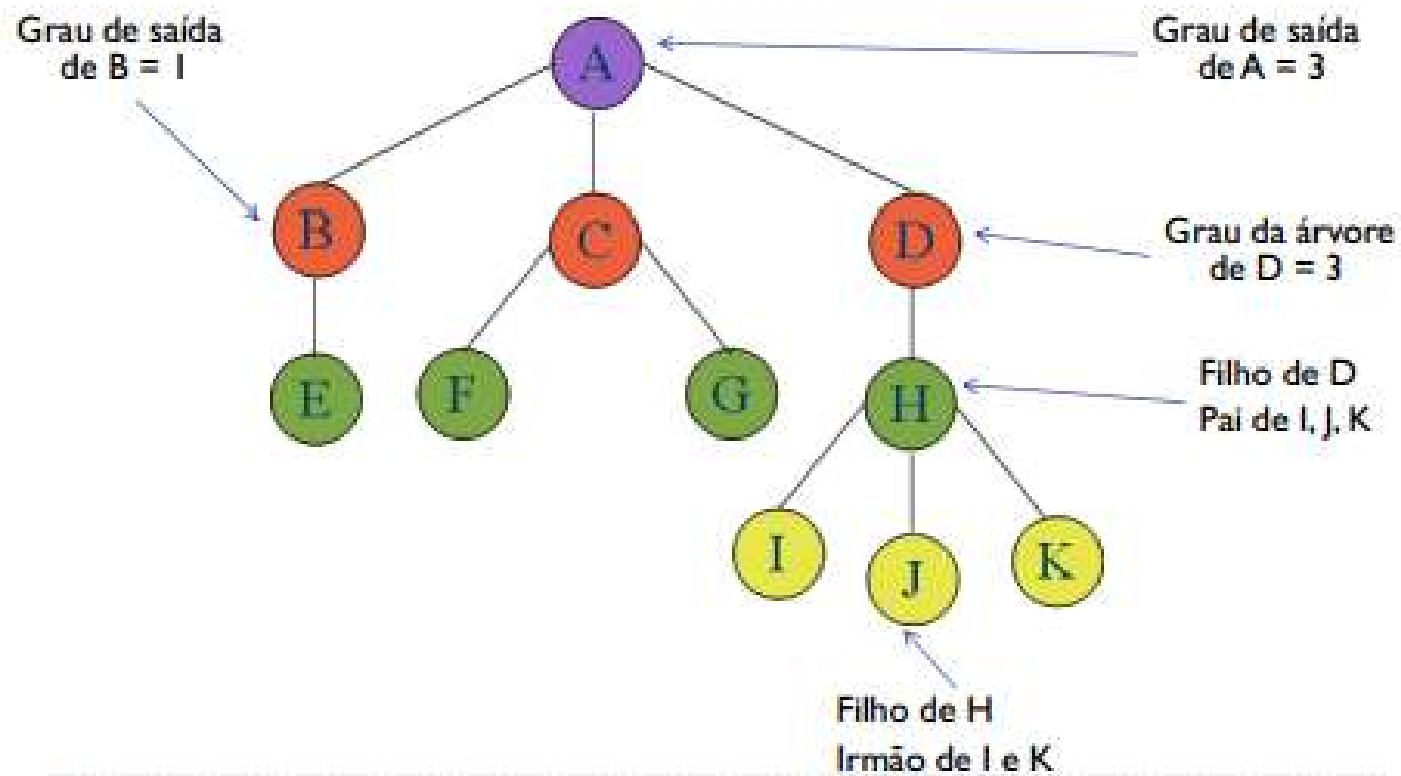


Árvores

■ Conceitos:

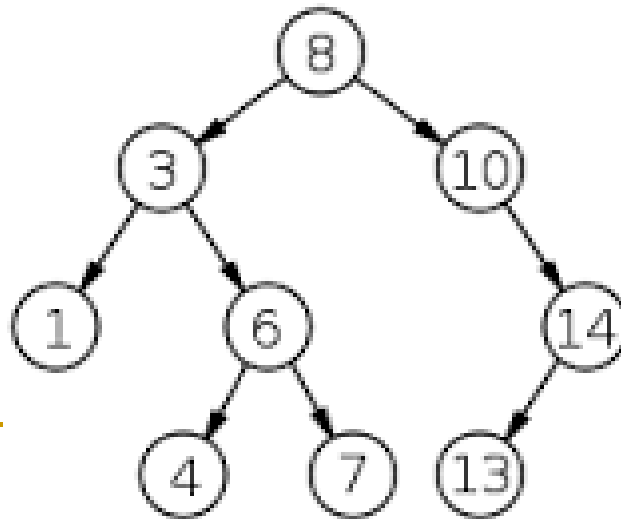
- Sejam T uma árvore e v um nó, tal que $v \in T$. T_v é a sub-árvore de T que tem v como raiz. Definem-se as seguintes propriedades:
 - O grau de saída do nó v é o número de sub-árvores que ele possui;
 - O grau da árvore T é o maior grau de saída dentre os de todos os seus nós;
 - os filhos de v são as raízes das sub-árvores de v ;
 - se w_1 e w_2 são filhos de v então v é o pai de w_1 e w_2 e estes são irmãos;
-

Árvores



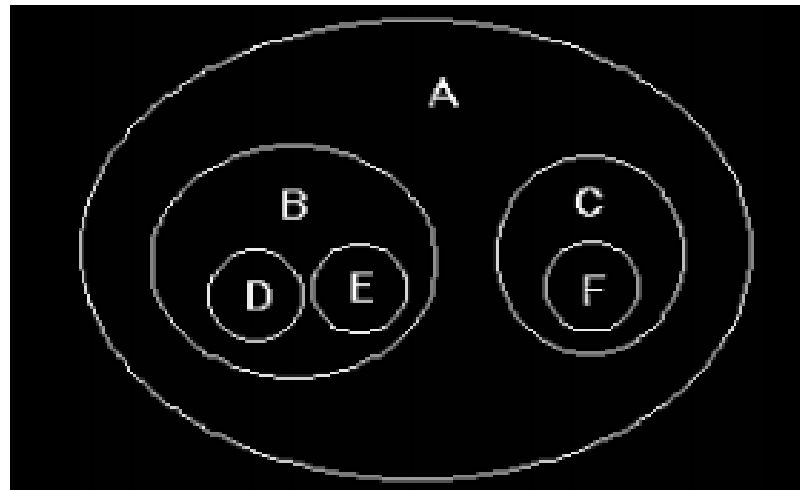
Árvores Binárias

- Uma **árvore binária** é uma ED caracterizada por:
 - Não ter elemento algum (árvore vazia).
 - Tem um elemento distinto, denominado raiz, com dois ponteiros para duas estruturas diferentes, denominadas sub-árvore esquerda e sub-árvore direita.



Árvores Binárias

- Conteúdo do nó pode ser de qualquer tipo.
- Esquemáticamente, uma árvore pode ser representada da seguinte maneira:

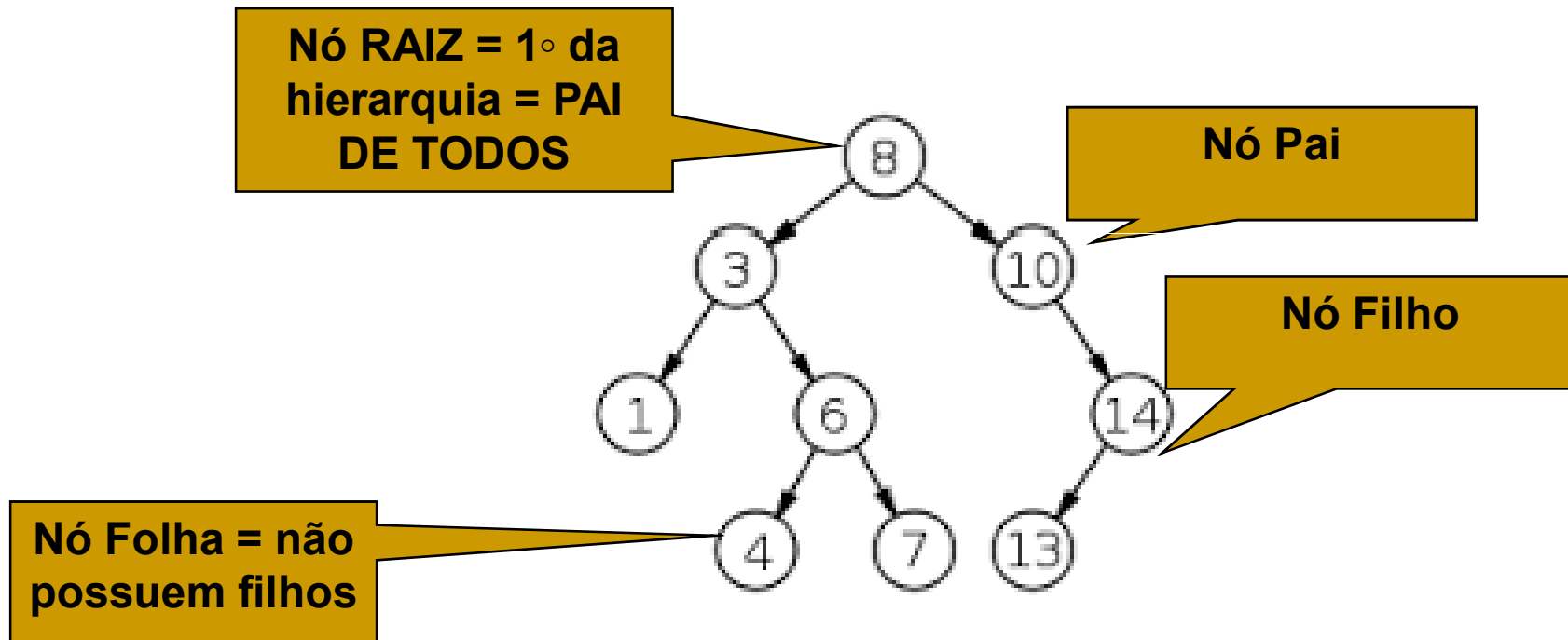


Árvores Binárias

- A definição é recursiva e, devido a isso, muitas operações sobre árvores binárias utilizam recursão.
 - É o tipo de árvore mais utilizado na computação.
 - A principal utilização de árvores binárias são as árvores de busca binária.
-

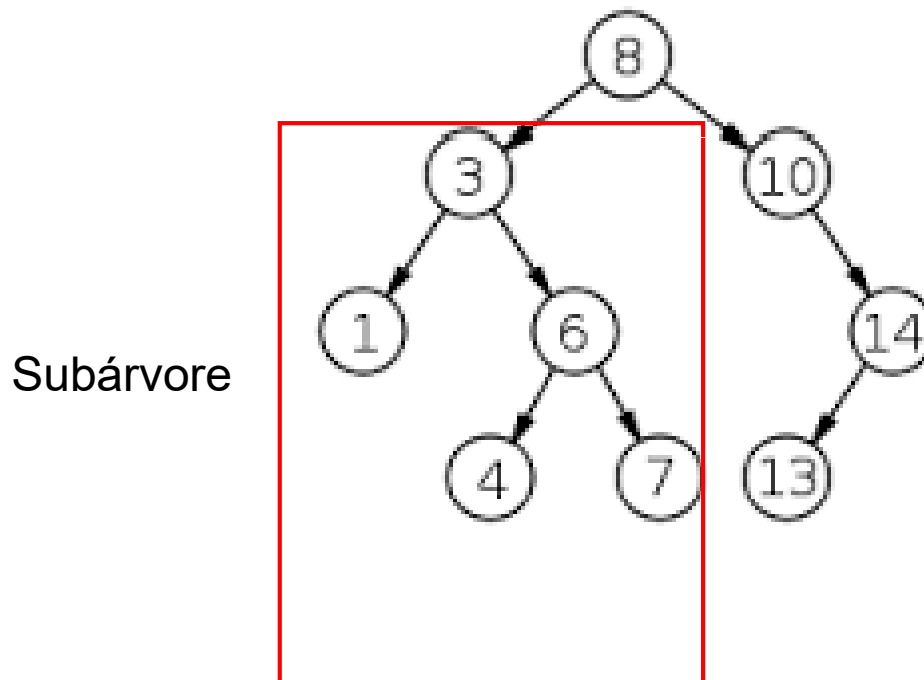
Árvores Binárias

■ Tipos de Nó



Árvores Binárias

- Esquemáticamente, uma árvore pode ser representada da seguinte maneira:



Árvores Binárias

- Representação por expressão com parênteses aninhados.
 - Cada conjunto de parênteses correspondentes contém um nodo e seus filhos.
 - Se um nodo não tem filhos, ele é seguido por um par de parênteses sem conteúdo.

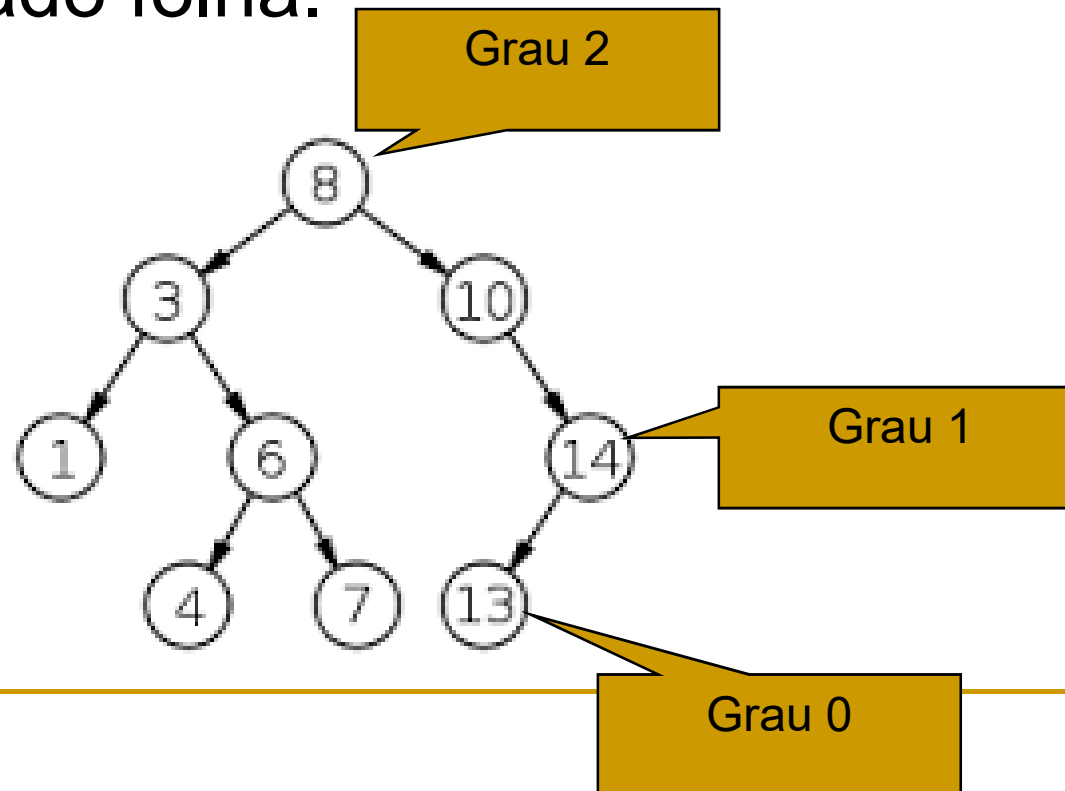
A (B (D () E ()) C (F ()))

Árvores Binárias

- A primeira representações permitem uma melhor visualização das árvores.
 - A última, por sua vez, facilitam a persistência dos nodos das árvores (em arquivos, por exemplo), possibilitando assim a sua reconstituição.
-

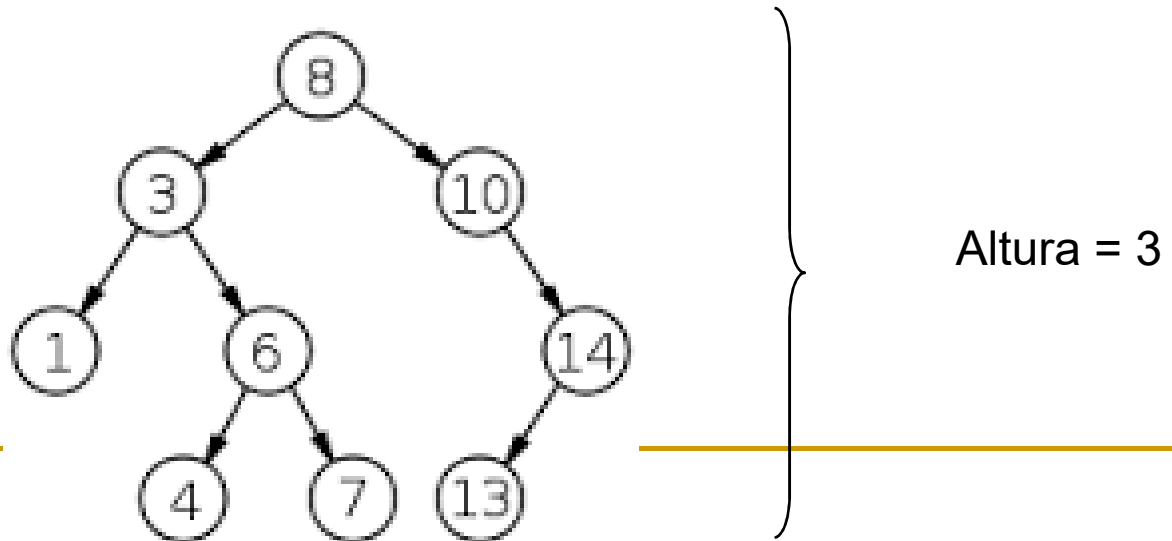
Árvores Binárias

- Os nós de uma árvore binária possuem graus zero, um ou dois. Um nó de grau zero é denominado folha.



Árvores Binárias

- Em uma árvore binária, por definição, cada nó poderá ter até duas folhas.
- A **profundidade** de um nó é a distância deste nó até a raiz. A maior profundidade de um nó, é a **altura** da árvore.



Árvores Binárias

■ Caminhamento:

- Formas de percorrer ou caminhar em uma árvore visitando seus nós:
 - Pré-ordem (Pré-fixada)
 - Central (Infixada)
 - Pós-ordem (Pós-fixada)

 - Para todas elas:
 - Se T é uma árvore nula, então não há nó a ser visitado.
 - Se T é uma árvore de um único nó então a visita ocorre apenas neste nó.
-

Árvores Binárias

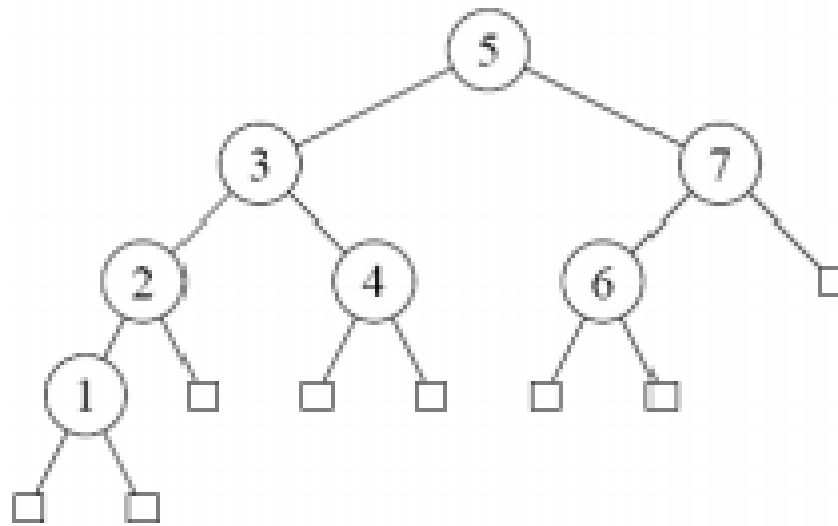
■ Caminhamento Pré-ordem:

- Visita o nó raiz, seguido de suas subárvores (da esquerda para a direita), cada uma em pré-ordem.

```
void preOrdem(Struct No *pNo) {  
    if (pNo != NULL) {  
        visita(pNo) ;  
        preOrdem(pNo→pEsquerda) ;  
        preOrdem(pNo→pDireita) ;  
    }  
}
```

Árvores Binárias

- Exemplo caminhamento pré-ordem:



- 5, 3 , 2, 1, 4, 7 e 6

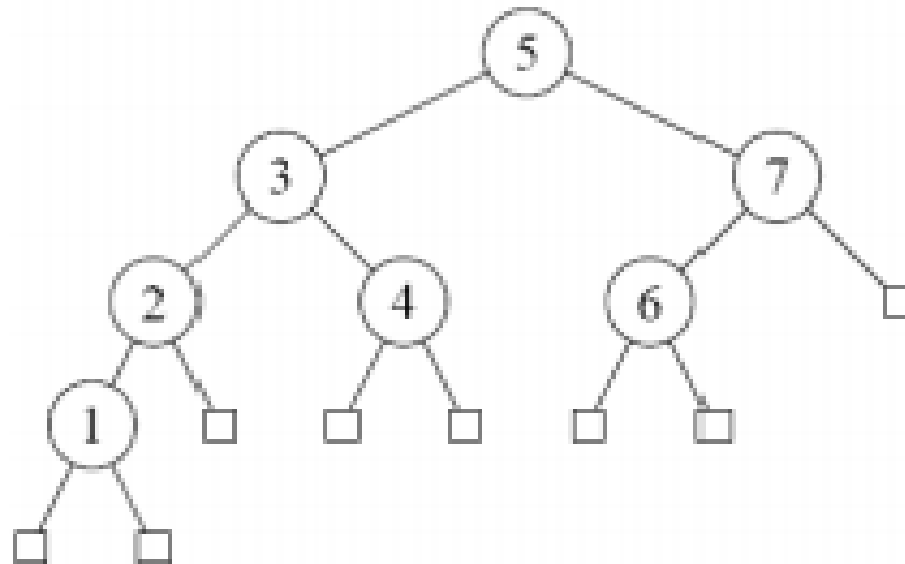
Árvores Binárias

- Caminhamento Central ou Em-ordem:
 - Visita os nós da 1ª subárvore à esquerda usando o caminhamento central, visita o nó raiz, visita as demais subárvores (a partir da 2ª) em caminhamento central (da esquerda para a direita).

```
void emOrdem(struct No *pNo) {      if (pNo !=  
    NULL) {  
        emOrdem(pNo→pEsquerda) ;  
        visita (pNo) ;  
        emOrdem(pNo→pDireita) ;  
    } }
```

Árvores Binárias

- Exemplo caminhamento Em-ordem ou central:



- 1,2,3,4,5,6 e 7

Árvores Binárias

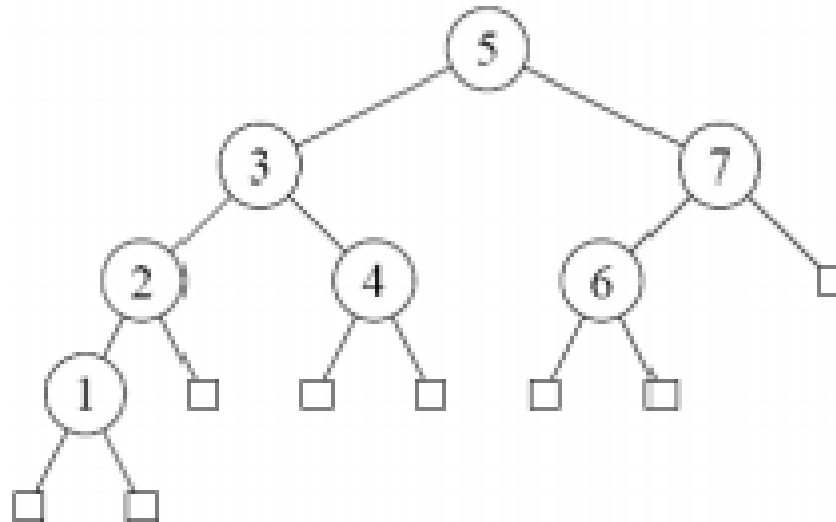
■ Caminhamento Pós-ordem:

- Visita o nodos das subárvores (da esquerda para a direita) cada uma em pós-ordem, visita o nó raiz.

```
void posOrdem(struct No *pNo) {  
    if (pNo != NULL) {  
        emOrdem(pNo→pEsquerda) ;  
        emOrdem(pNo→pDireita) ;  
        visita(pNo) ;  
    }  
}
```

Árvores Binárias

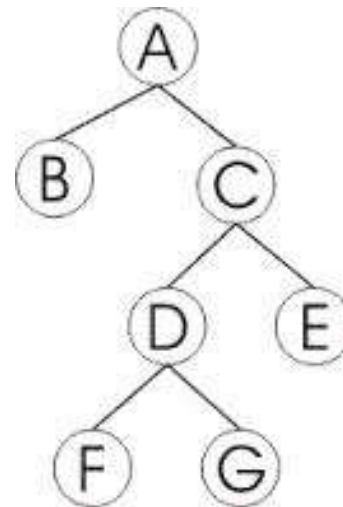
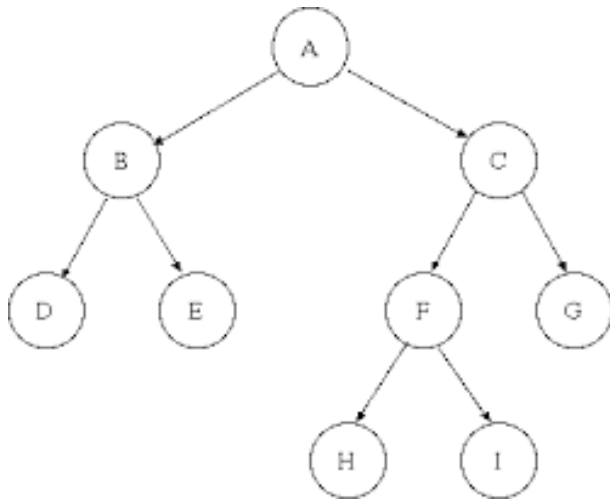
- Exemplo caminhamento Pós-ordem:



- 1, 2, 4, 3, 6, 7 e 5

Árvores Binárias

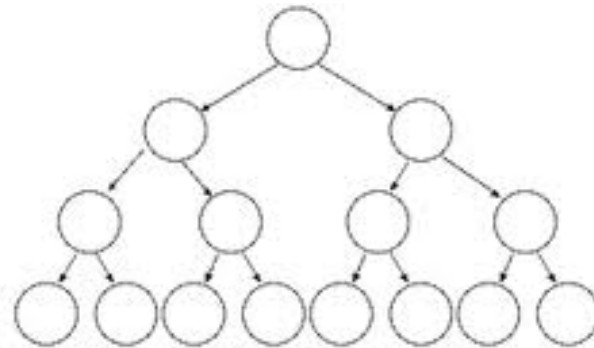
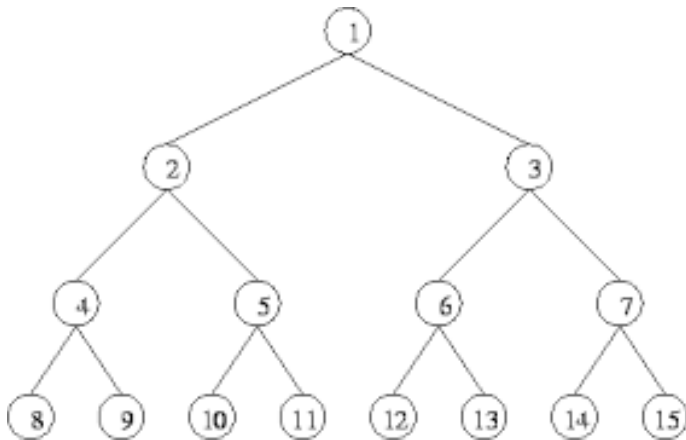
- **Árvore Estritamente Binária**
 - Se cada nó não-folha em uma árvore binária não tem subárvores esquerda e direita vazias.



Árvores Binárias

■ Árvore Binária Completa

- Uma árvore binária completa de nível n é a árvore estritamente binária, onde todos os nós folhas estão no nível n .



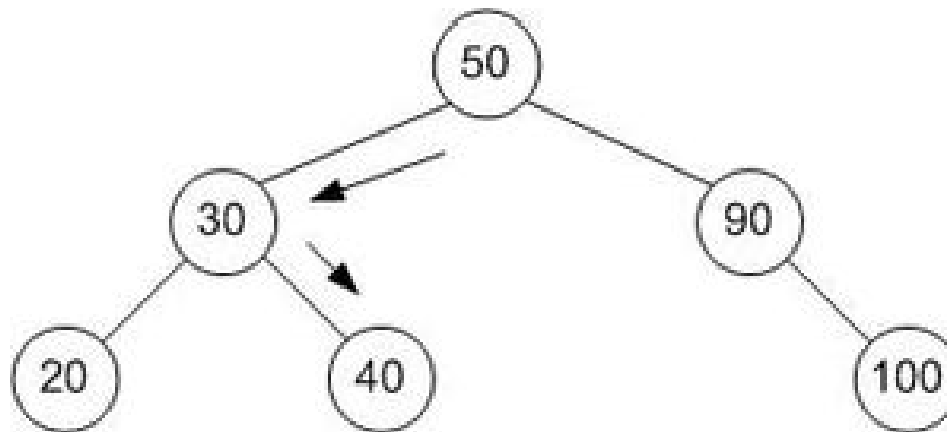
Árvores Binárias de Busca

■ Procedimento Busca

- Para encontrar um registro com uma chave x :
 - Compare-a com a chave que está na raiz.
 - Se x é menor, vá para a subárvore esquerda.
 - Se x é maior, vá para a subárvore direita.
 - Repita o processo recursivamente, até que a chave procurada seja encontrada ou um nó folha é atingido.
 - Se a pesquisa tiver sucesso então o conteúdo do registro retorna no próprio registro x .
-

Árvores Binárias

- Existe elemento 40?



Árvores Binárias

■ Inserção

□ Onde inserir?

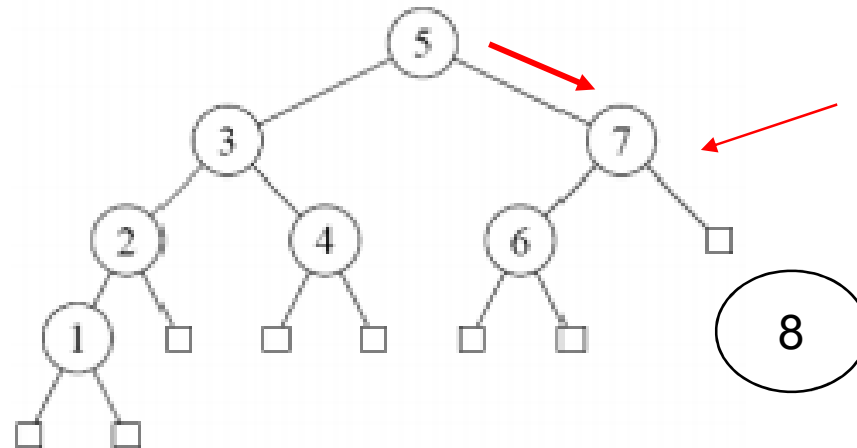
- Atingir um apontador nulo em um processo de pesquisa significa uma pesquisa sem sucesso.
- O apontador nulo atingido é o ponto de inserção.

□ Como inserir?

- Cria célula contendo registro (nó)
 - Procura lugar na árvore
 - Se registro não tiver na árvore, insere-o
-

Árvores Binárias

- Inserir o nó com chave 8...



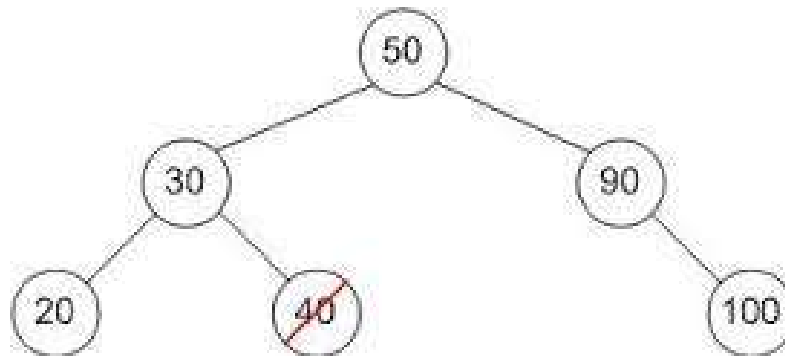
Árvore Binária

- Remoção: Existem três situações possíveis:
 - A primeira é quando se deseja retirar um elemento que é folha da árvore.
 - A segunda situação acontece quando o nó a ser retirado possui um único filho.
 - A terceira situação ocorre quando o nó a ser retirado tem dois filhos.
-

Árvore Binária

■ Remoção:

- A primeira, e mais simples, é quando se deseja retirar um elemento que é folha da árvore. Neste caso, basta retirar o elemento da árvore e atualizar o pai, pois seu filho não existe mais.

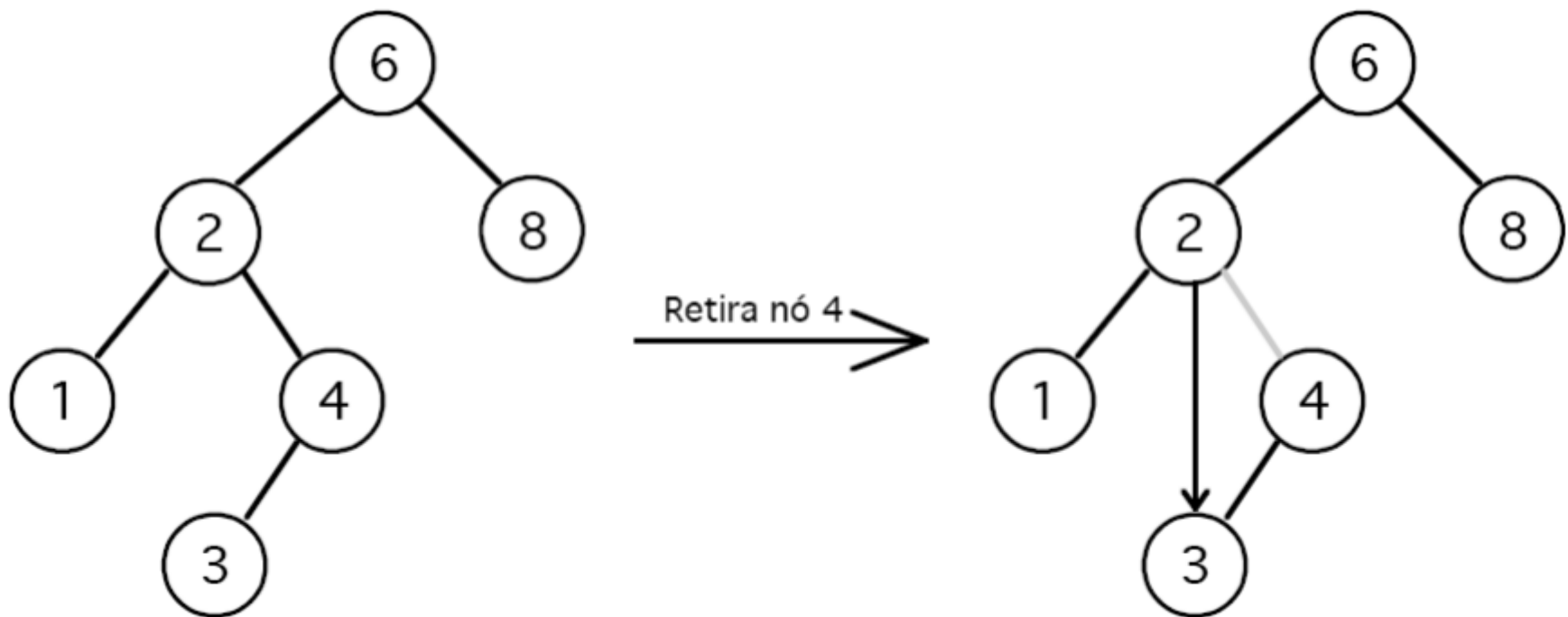


Árvore Binária

■ Remoção:

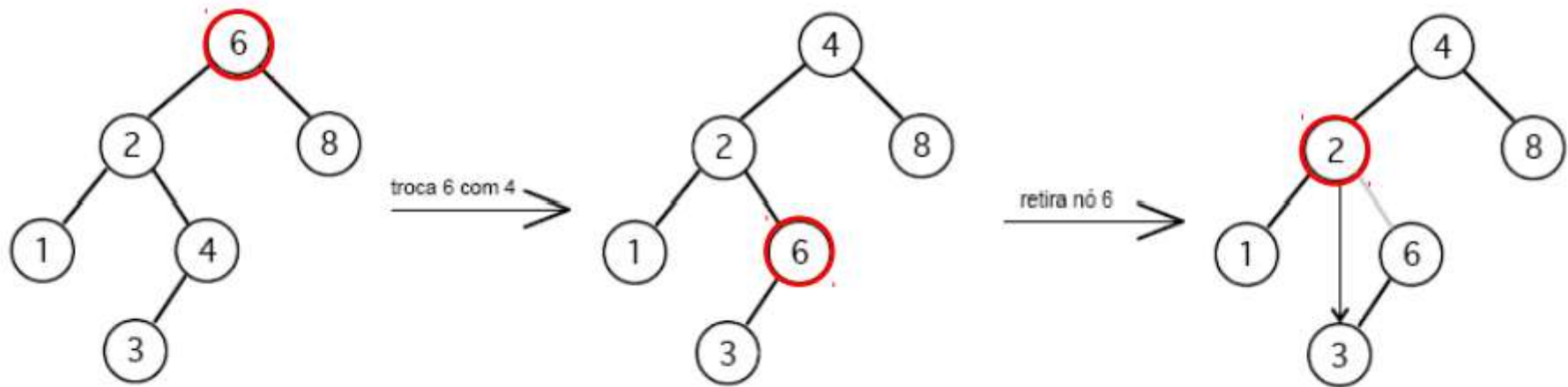
- A segunda situação, ainda simples, acontece quando o nó a ser retirado possui um único filho. Para retirar esse elemento é necessário antes acertar o ponteiro do pai, “pulando” o nó: o único neto passa a ser filho direto.
-

Árvore Binária



Árvore Binária

■ Terceira Situação:

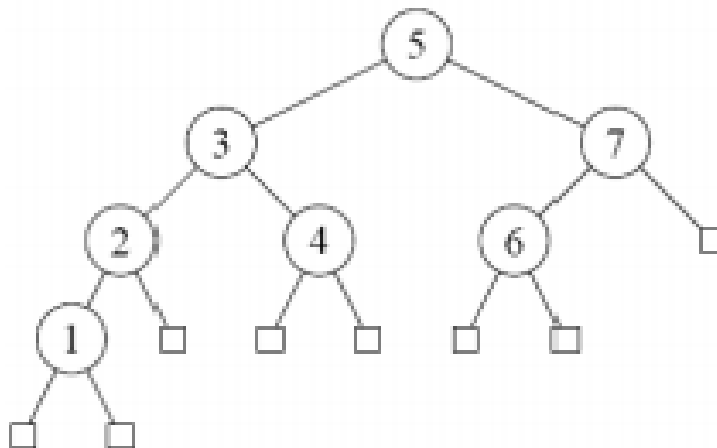


- ❑ Localiza o nó a ser eliminado (nó 6)
- ❑ Encontra o nó mais a direita da sub-árvore da esquerda (nó 4)
- ❑ Trocar os valores entre o nó a ser eliminado e o nó encontrado.
- ❑ Localiza novamente o nó a ser eliminado (nó 6)

Árvores Binárias

■ Como retirar?

- ❑ Folha?
- ❑ Nó interno, contendo um filho?
- ❑ Raiz da árvore ou subárvore?

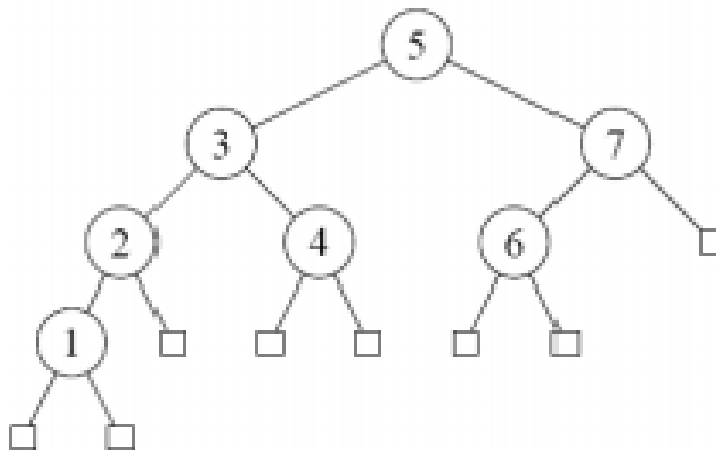


Árvores Binárias

- Como retirar?
 - Se o nó que contém o registro a ser retirado possui, no máximo, um descendente a operação é menos trabalhosa.
 - No caso do nó conter dois descendentes, o registro a ser retirado deve ser primeiro:
 - ❑ substituído pelo registro mais à direita na subárvore esquerda;
 - ❑ ou pelo registro mais à esquerda na subárvore direita.
-

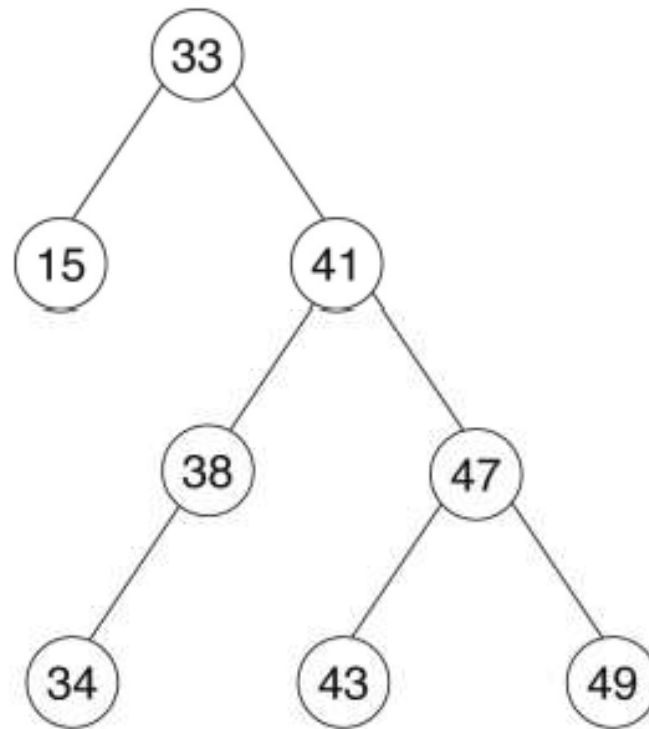
Árvores Binárias

■ Como retirar?



- Assim: para retirar o registro com chave 5 da árvore basta trocá-lo pelo registro com chave 4 ou pelo registro com chave 6, e então retirar o nó que recebeu o registro com chave 5.

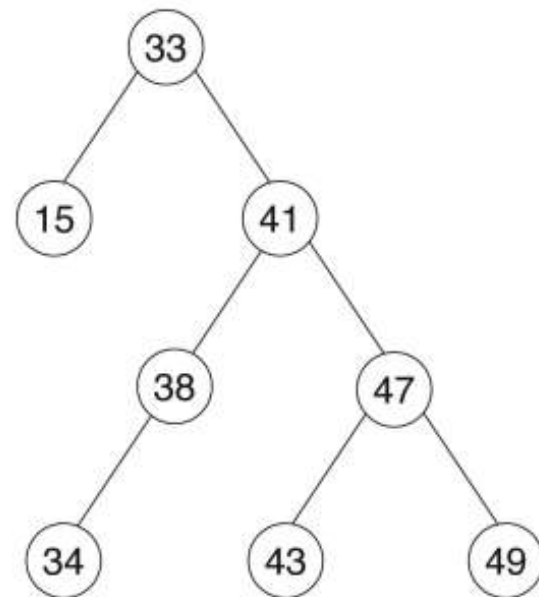
Exercício



Exercício

- Tendo como base a árvore, faça o que se pede nos itens a seguir.
 - Considerando que o nó de valor 33 seja a raiz da árvore, descreva a ordem de visita para uma varredura em in-ordem (in-fixado) na árvore

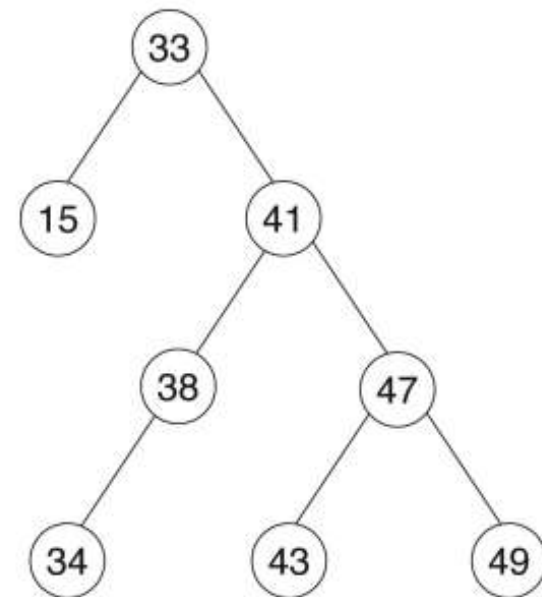
15, 33, 34, 38, 41, 47, 49



Exercício

- Tendo como base a árvore acima, faça o que se pede nos itens a seguir.
 - Considerando que o nó de valor 33 seja a raiz da árvore, descreva a ordem de visita para uma varredura em pré-ordem (r-e-d, ou pré-fixado à esquerda) na árvore.

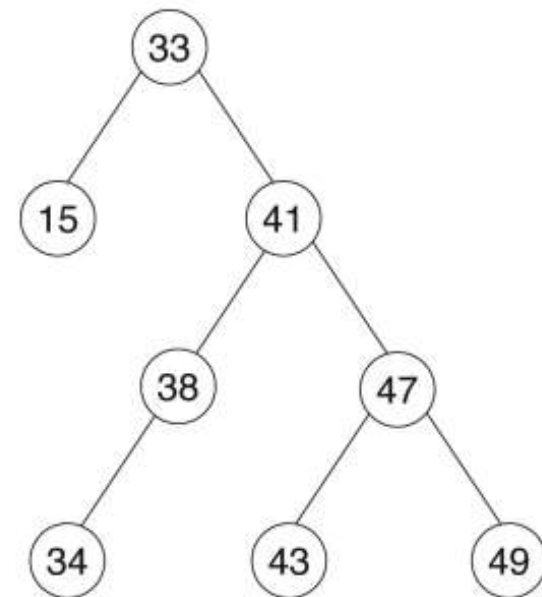
33, 15, 41, 38, 34, 47, 43, 49



Exercício

- Tendo como base a árvore acima, faça o que se pede nos itens a seguir.
 - Considerando que o nó de valor 33 seja a raiz da árvore, descreva a ordem de visita para uma varredura em pós-ordem (pós-fixado) na árvore.

15, 34, 38, 43, 49, 47, 41, 33

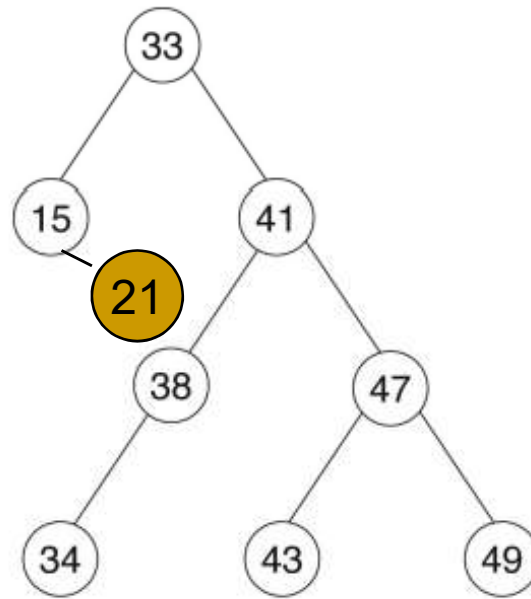


Exercício

- ❑ Considerando que a árvore cuja raiz é o nó de valor 33 represente uma árvore de busca binária, desenhe a nova árvore que será obtida após a realização das seguintes operações: inserir um nó de valor 21; remover o nó de valor 47; inserir um nó de valor 48.
-

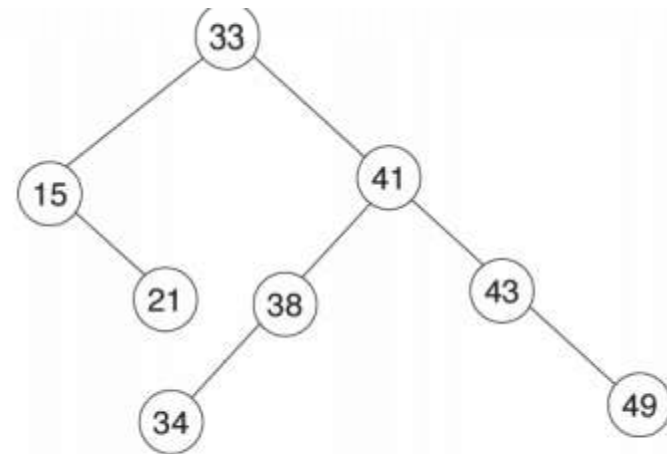
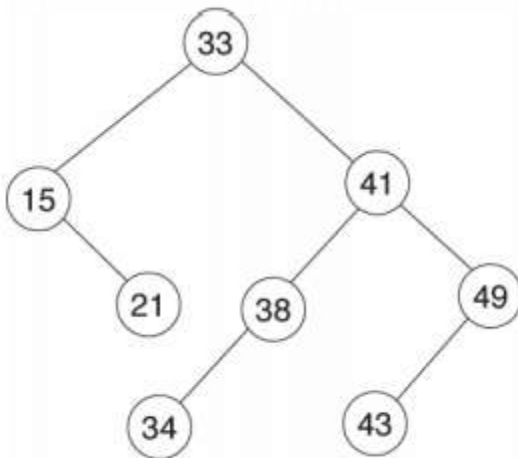
Exercício

- inserir um nó de valor 21



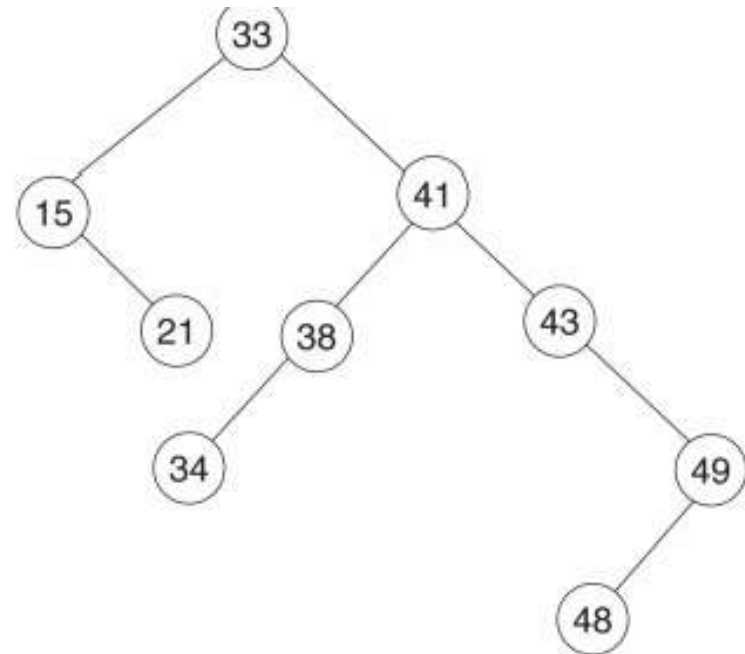
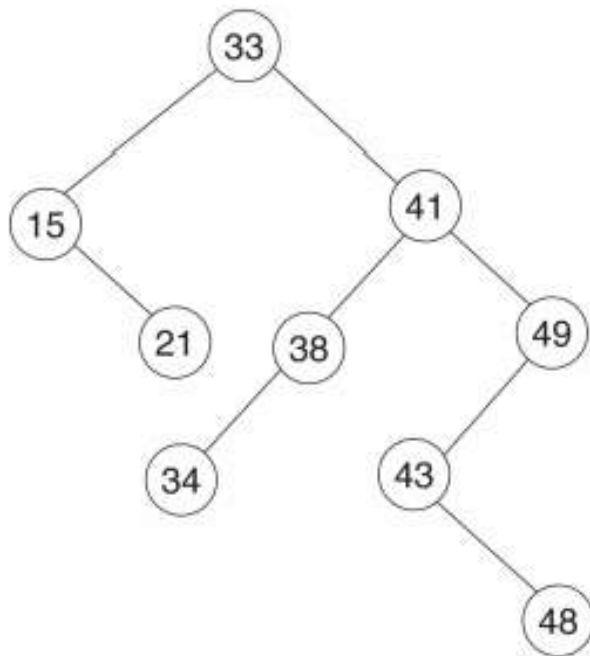
Exercício

- remover o nó de valor 47: há duas possibilidades: um dos nós 43 ou 49 deve ser “promovido” para o lugar do nó 47, resultando em uma das figuras abaixo.



Exercício

- inserir um nó de valor 48.



Exercício
