



UNIVERSIDADE FEDERAL DE ITAJUBÁ

Algoritmos e Estrutura de Dados I

COM 111

Árvore Binária de Pesquisa

Vanessa Cristina Oliveira de Souza



Introdução

- ▶ A árvore de pesquisa é uma **estrutura de dados** muito eficiente para armazenar e recuperar informação.
- ▶ Uma árvore binária de busca serve para o armazenamento de dados na memória principal do computador e a sua subsequente recuperação.





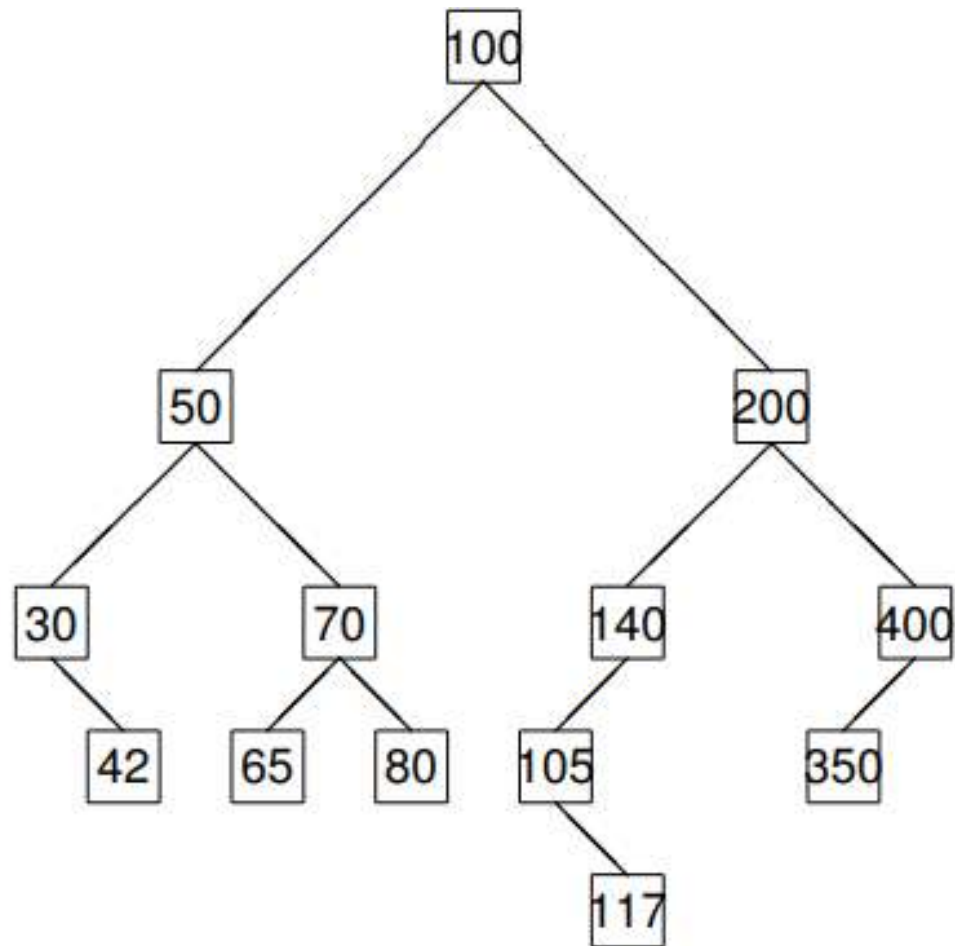
Introdução

- ▶ Uma árvore binária é definida como um conjunto finito de **nós** que ou está **vazio** ou consiste de um nó chamado **raiz** mais os elementos de duas árvores binárias distintas chamadas de **subárvores esquerda e direita** do nó raiz.
- ▶ Em uma árvore binária:
 - ▶ Cada nó possui uma chave
 - ▶ Cada nó tem no máximo duas subárvores.





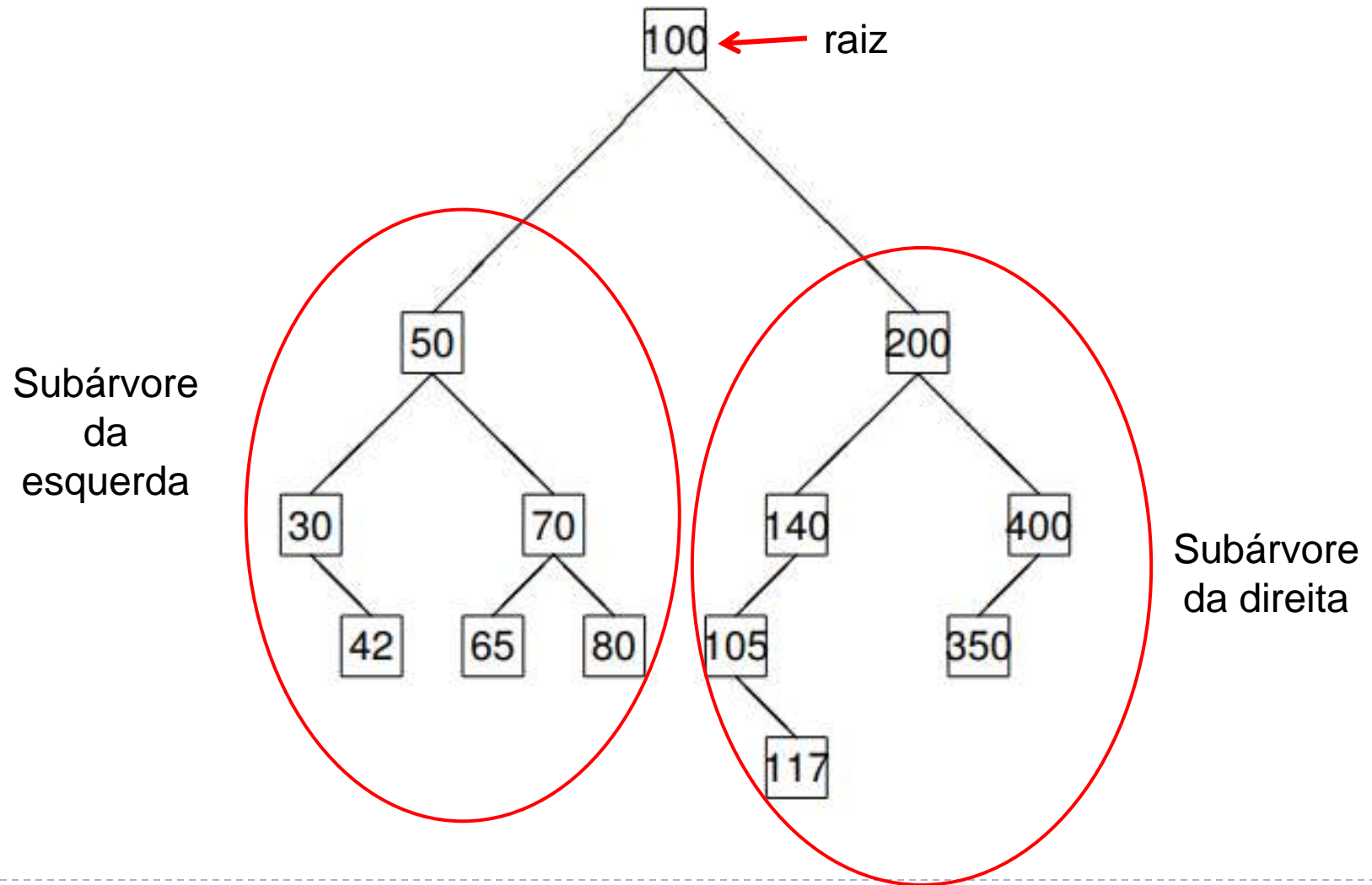
Introdução



Propriedades da Árvore Binária

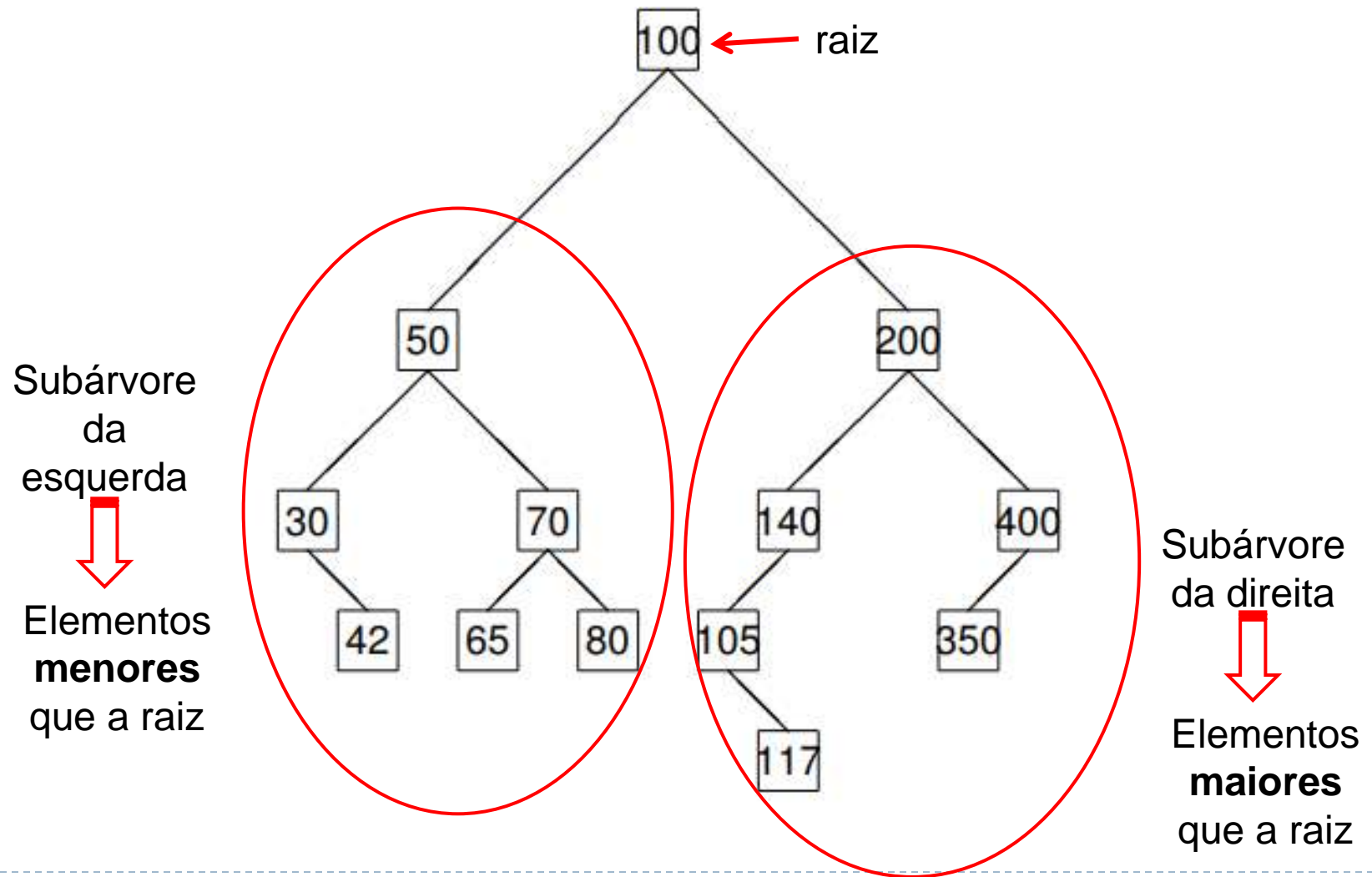


Propriedades





Propriedades





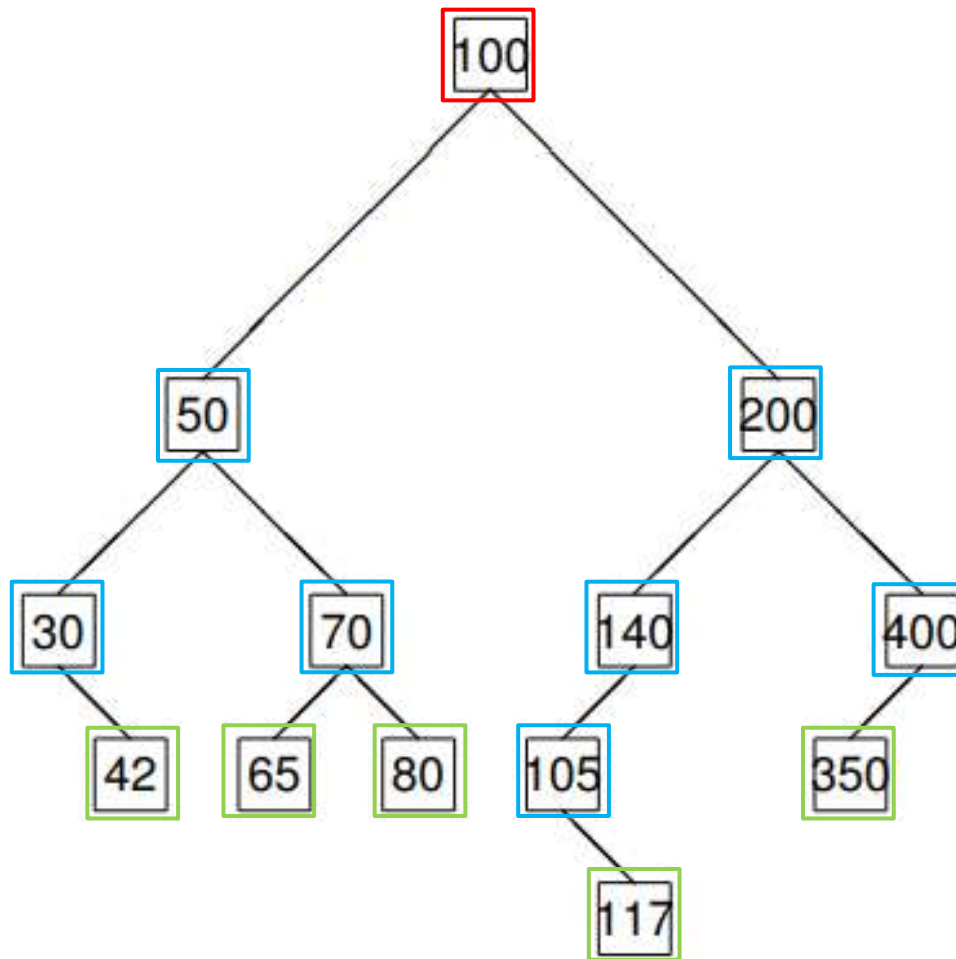
Propriedades

- ▶ O número de subárvores de um nó é chamado de grau daquele nó:
 - ▶ Nó de grau zero
 - ▶ Não tem filhos
 - ▶ É chamado de nó externo ou folha
 - ▶ Nó de grau 1
 - ▶ Possui apenas um filho (ou o da esquerda, ou o da direita)
 - ▶ Nó de grau 2
 - ▶ Possui os dois filhos (esquerda e direita)





Propriedades



- Nó raiz
- Nós Intermediários
- Nós folha





Propriedades

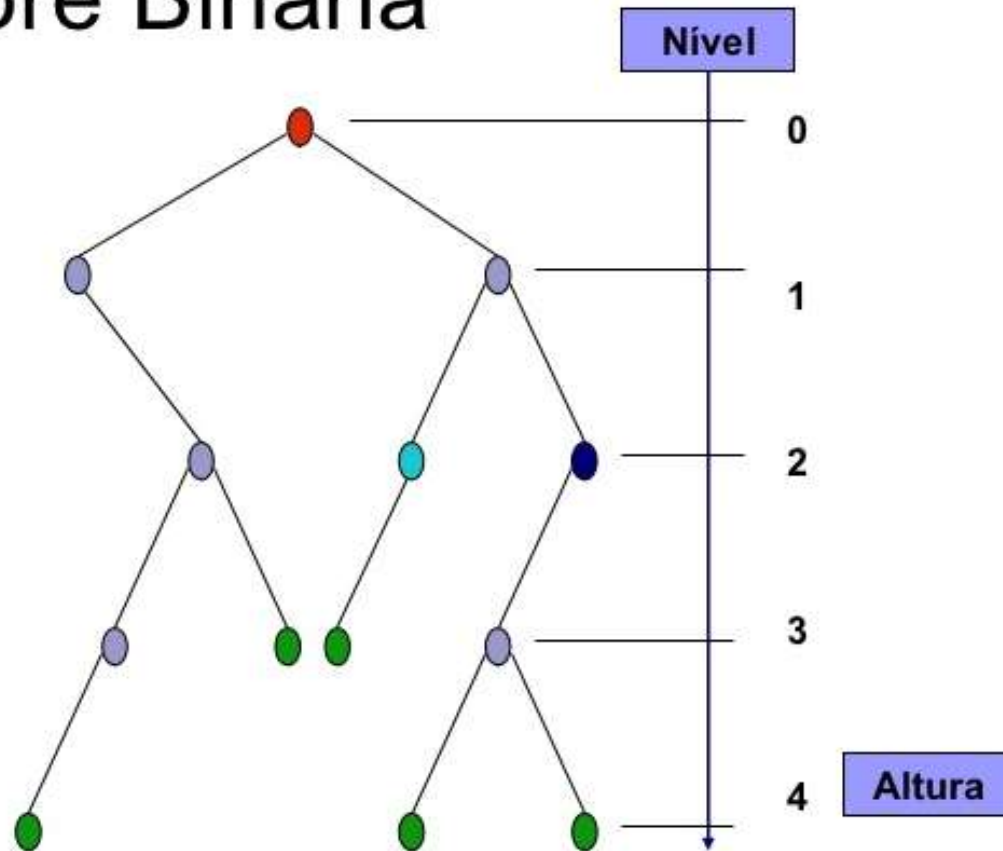
- ▶ **Altura da Árvore Binária**
 - ▶ Comprimento do caminho mais longo da raiz até uma das folhas
 - ▶ A altura da raiz é **zero**
 - ▶ A altura de uma árvore vazia é -1
 - ▶ Determina o **esforço computacional** para encontrar um determinado nó





Propriedades

Árvore Binária



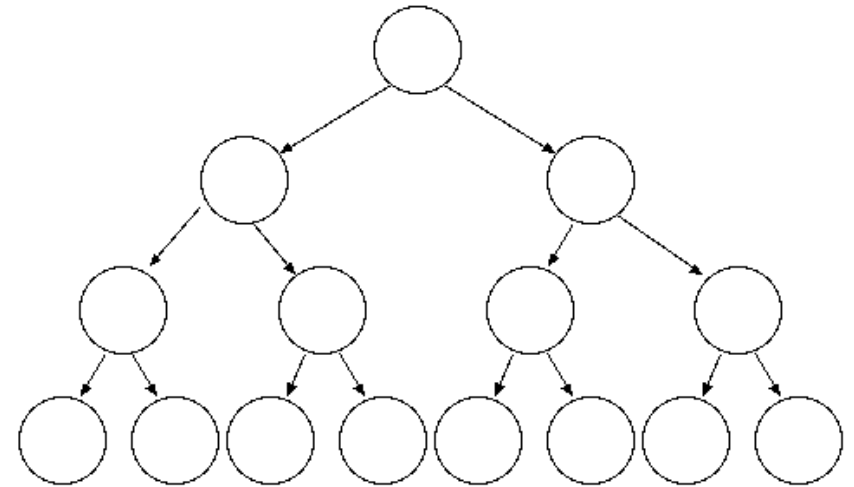
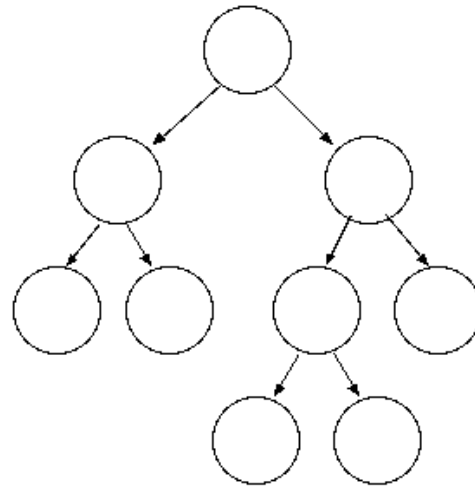
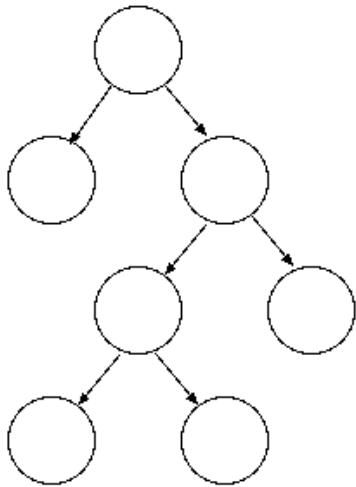
Classificação de Árvores Binárias



Classificação de Árvores Binárias

- ▶ Uma árvore binária pode ser:
 - ▶ Estritamente binária
 - ▶ Seus nós são de grau 0 ou grau 2
 - ▶ Completa
 - ▶ Se existem nós de grau 0, ou eles estão no penúltimo, ou no último nível
 - ▶ Cheia
 - ▶ Se existem nós de grau 0, eles estão no último nível da árvore



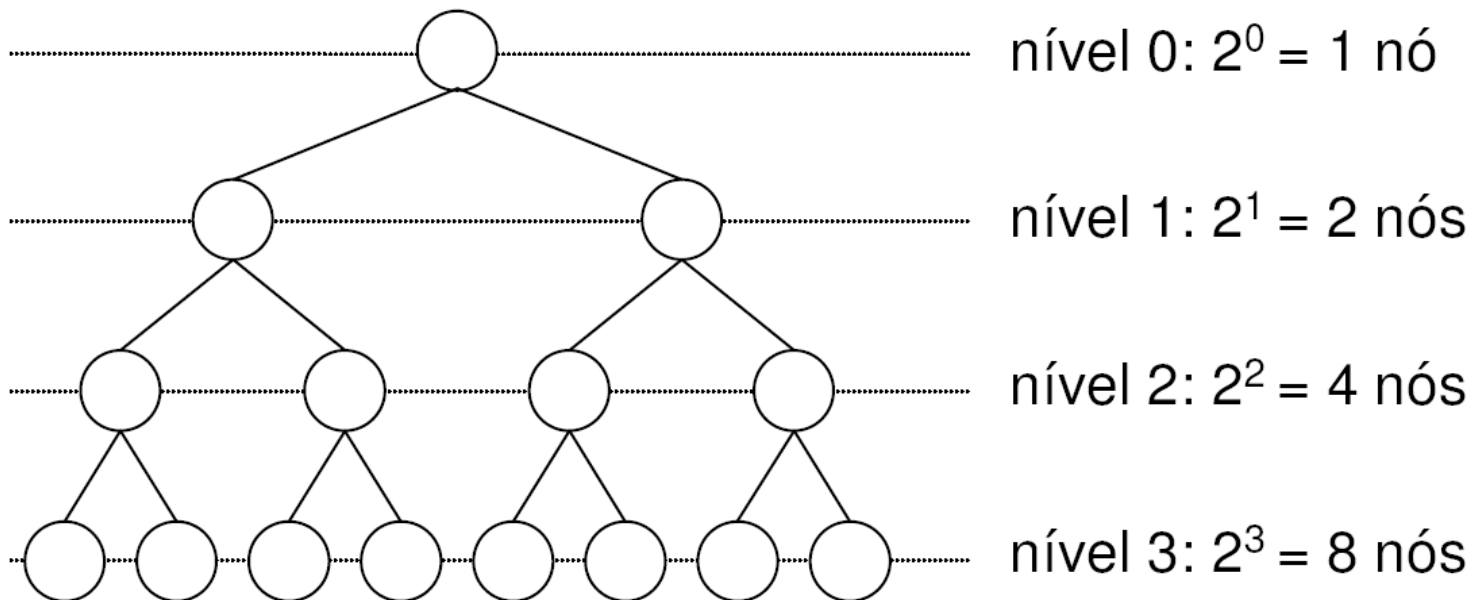




Classificação de Árvores Binárias

▶ **Árvore cheia**

- ▶ É uma árvore completamente balanceada.
- ▶ Complexidade $O(\log_2(n))$

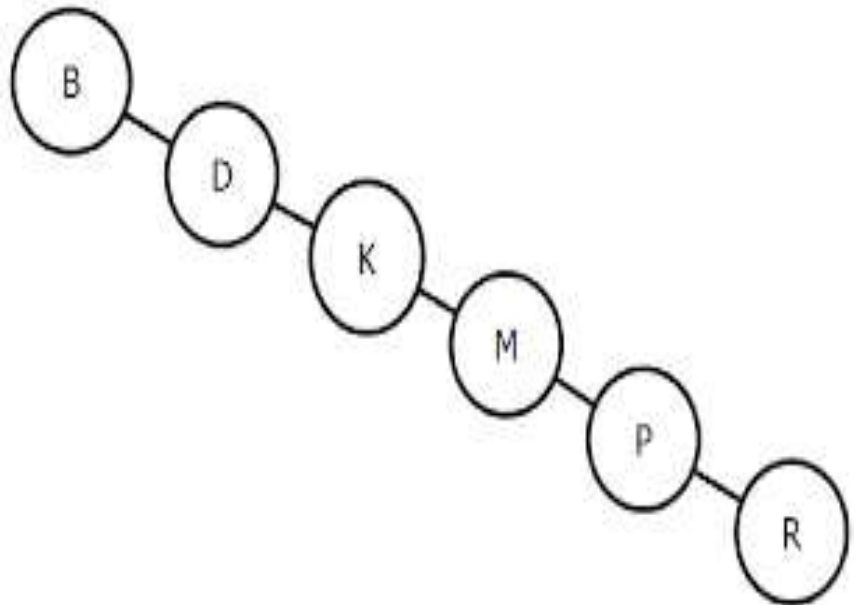




Classificação de Árvores Binárias

▶ **Árvore Degenerada**

- ▶ É uma árvore completamente **desbalanceada**, ou seja, torna-se uma **lista**.
- ▶ A altura da árvore é n
- ▶ Complexidade $O(n)$



Estrutura do TAD Árvore Binária de Pesquisa

TAD ABP



Estrutura

- ▶ O nó da árvore binária possui a chave e dois ponteiros : um para a subárvore esquerda e outro para a subárvore direita

```
struct noArvore
{
    int chave;
    struct noArvore *esq;
    struct noArvore *dir;
}
```



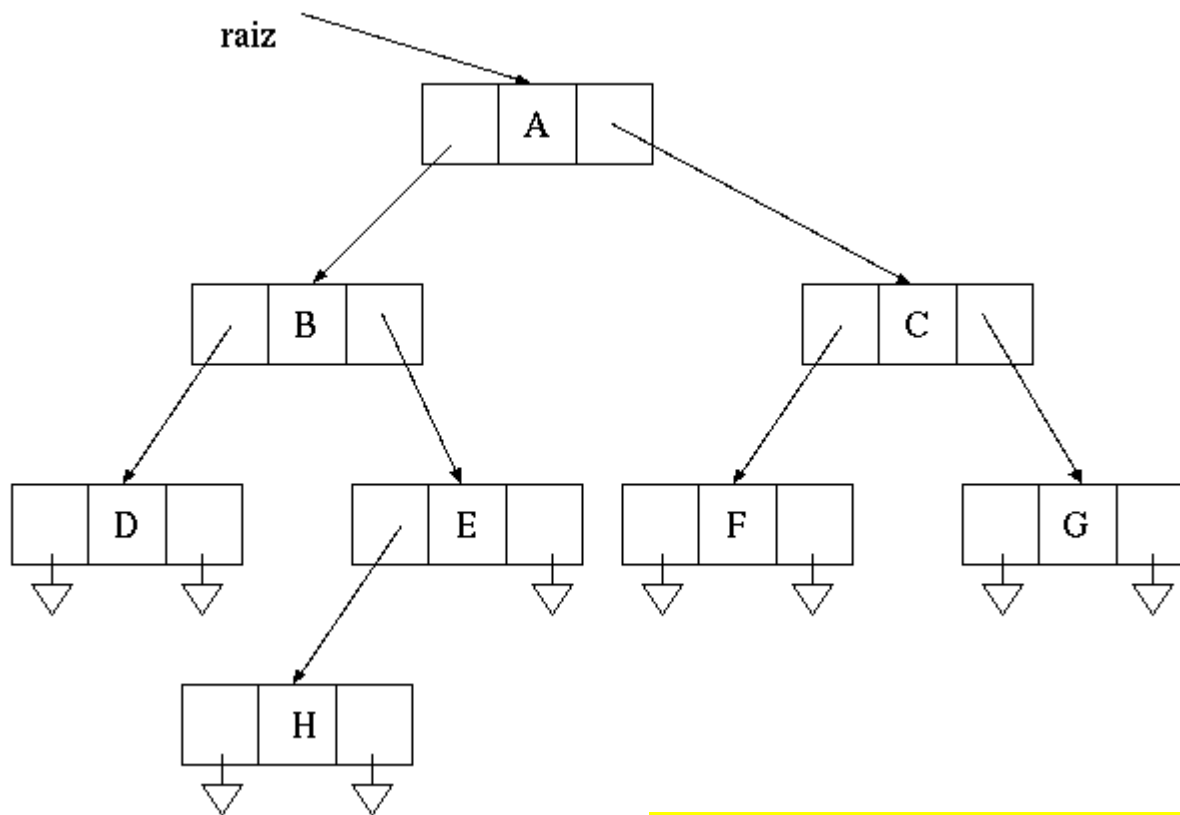
Estrutura

- ▶ O tipo árvore é identificado pelo seu nó raiz
 - ▶ Se a árvore está vazia, a raiz é NULL

```
struct arvore
{
    struct noArvore *raiz;
}
```



Estrutura



A árvore está errada!!!!

Operações em Árvores Binárias



- ▶ As principais operações sobre as árvores binárias são:
 - ▶ Inserir um elemento
 - ▶ Excluir um elemento
 - ▶ Buscar um elemento
 - ▶ Encontrar o maior
 - ▶ Encontrar o menor
 - ▶ Buscar o elemento sucessor e o predecessor
 - ▶ Percorrimento
 - ▶ Em pré-ordem
 - ▶ Em ordem
 - ▶ Em pós-ordem

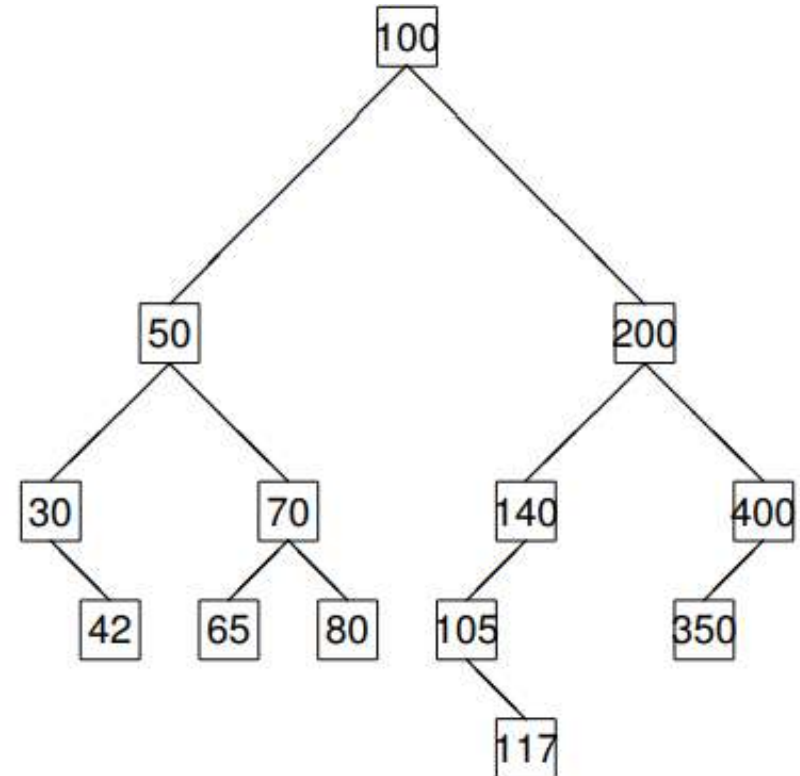


- ▶ As principais operações sobre as árvores binárias são:
 - ▶ Inserir um elemento
 - ▶ Excluir um elemento
 - ▶ **Buscar um elemento**
 - ▶ Encontrar o maior
 - ▶ Encontrar o menor
 - ▶ Buscar o elemento sucessor e o predecessor
 - ▶ Percorrimento
 - ▶ Em pré-ordem
 - ▶ Em ordem
 - ▶ Em pós-ordem



Buscar um elemento

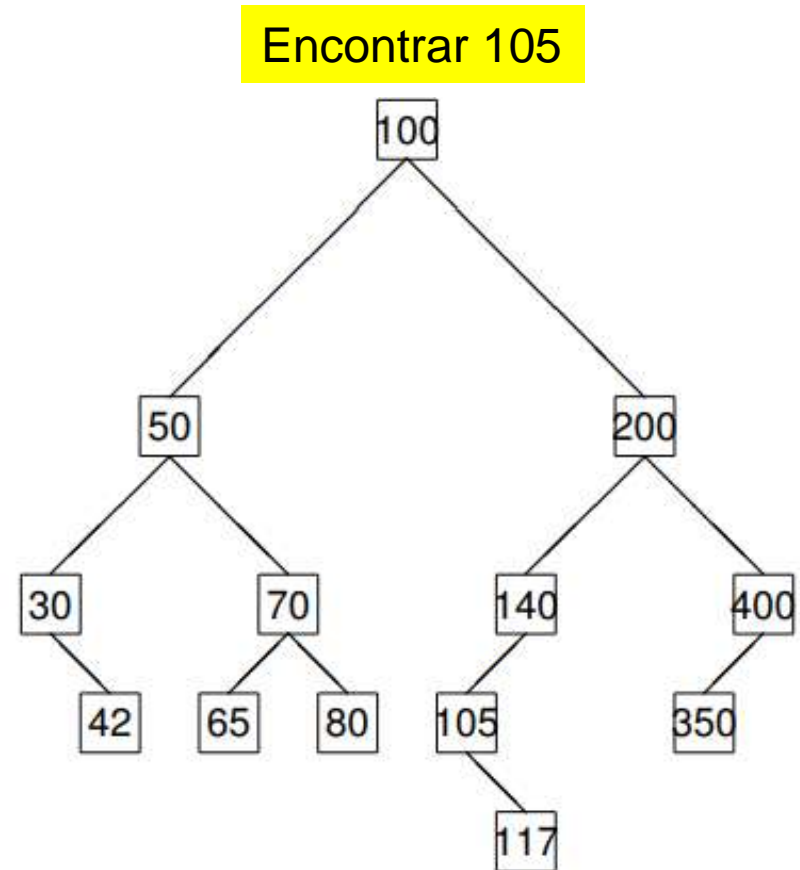
- ▶ A busca começa sempre na raiz
- ▶ Faz-se a comparação entre o elemento buscado e a chave
 - ▶ Se for igual:
 - ▶ Encontrou. Termina
 - ▶ Se o elemento é maior que a chave
 - ▶ Subárvore à direita
 - ▶ Se o elemento é menor que a chave
 - ▶ Subárvore à esquerda
- ▶ O processo para ao encontrar o elemento, ou encontrar NULL





Buscar um elemento

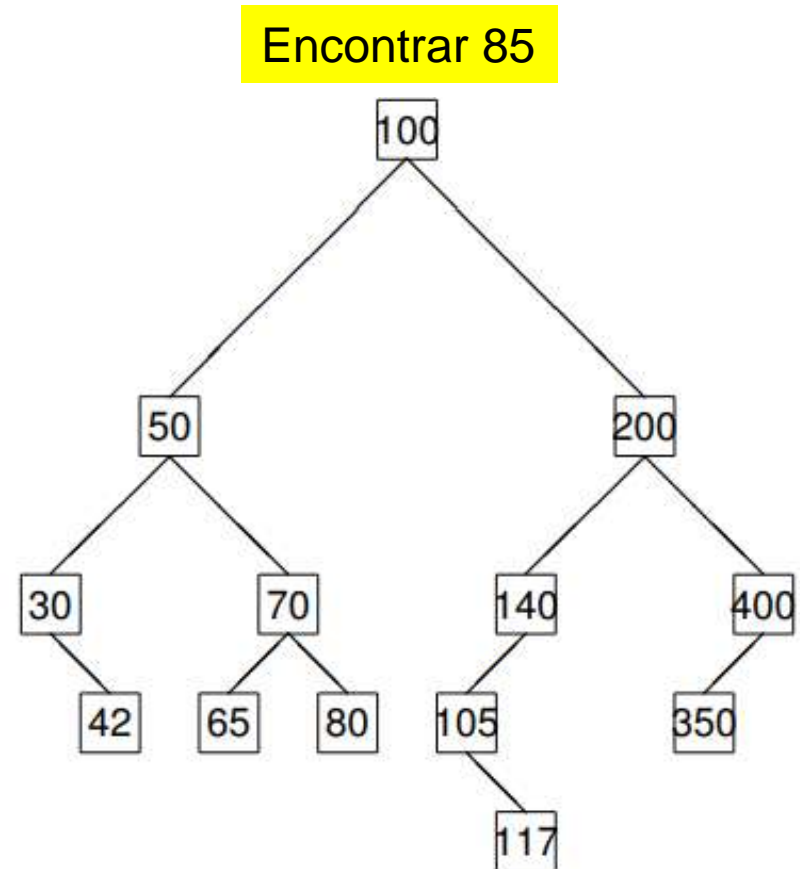
- ▶ A busca começa sempre na raiz
- ▶ Faz-se a comparação entre o elemento buscado e a chave
 - ▶ Se for igual:
 - ▶ Encontrou. Termina
 - ▶ Se o elemento é maior que a chave
 - ▶ Subárvore à direita
 - ▶ Se o elemento é menor que a chave
 - ▶ Subárvore à esquerda
- ▶ O processo para ao encontrar o elemento, ou encontrar NULL





Buscar um elemento

- ▶ A busca começa sempre na raiz
- ▶ Faz-se a comparação entre o elemento buscado e a chave
 - ▶ Se for igual:
 - ▶ Encontrou. Termina
 - ▶ Se o elemento é maior que a chave
 - ▶ Subárvore à direita
 - ▶ Se o elemento é menor que a chave
 - ▶ Subárvore à esquerda
- ▶ O processo para ao encontrar o elemento, ou encontrar NULL





- ▶ As principais operações sobre as árvores binárias são:
 - ▶ Inserir um elemento
 - ▶ Excluir um elemento
 - ▶ Buscar um elemento
 - ▶ Encontrar o maior
 - ▶ Encontrar o menor
 - ▶ Buscar o elemento sucessor e o predecessor
 - ▶ **Percorrimento**
 - ▶ Em pré-ordem
 - ▶ Em ordem
 - ▶ Em pós-ordem



Percorrimento em ABP

- ▶ Uma árvore é uma estrutura não sequencial.
- ▶ Sendo assim, não existe ordem natural para percorrer árvores e portanto podemos escolher diferentes maneiras de percorrê-las.
- ▶ Nós iremos estudar três métodos para percorrer árvores.
 - ▶ Pré-ordem
 - ▶ Em ordem
 - ▶ Pós-ordem





Percorrimento em ABP

▶ Pré-Ordem

- ▶ Raiz
- ▶ Esquerda
- ▶ Direita

▶ Em ordem

- ▶ Esquerda
- ▶ Raiz
- ▶ Direita

Todos estes três métodos podem ser definidos recursivamente

▶ Pós-Ordem

- ▶ Direita
- ▶ Raiz
- ▶ Esquerda





Percorrimento em ABP

▶ Pré-Ordem

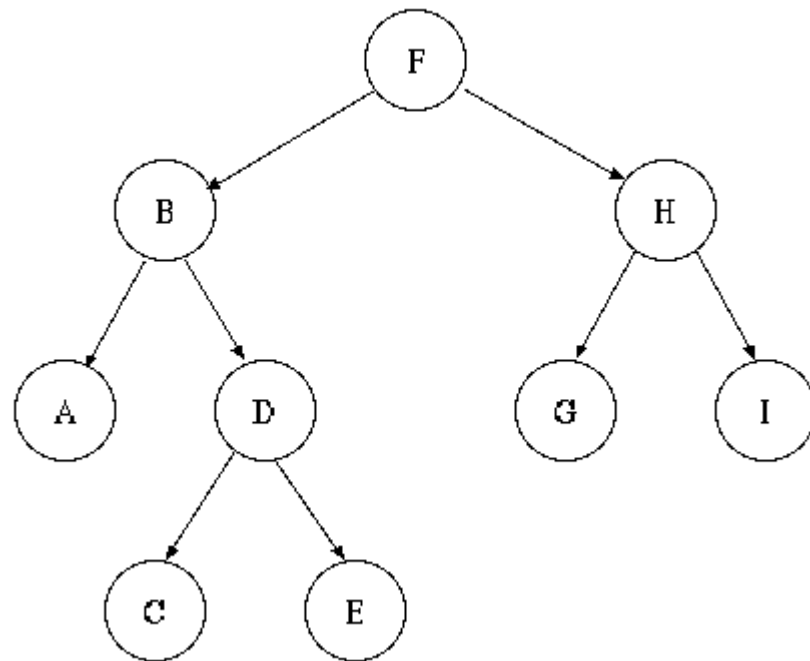
- ▶ Raiz
- ▶ Esquerda
- ▶ Direita

▶ Em ordem

- ▶ Esquerda
- ▶ Raiz
- ▶ Direita

▶ Pós-Ordem

- ▶ Direita
- ▶ Raiz
- ▶ Esquerda





- ▶ As principais operações sobre as árvores binárias são:
 - ▶ **Inserir um elemento**
 - ▶ Excluir um elemento
 - ▶ Buscar um elemento
 - ▶ Encontrar o maior
 - ▶ Encontrar o menor
 - ▶ Buscar o elemento sucessor e o predecessor
 - ▶ Percorrimento
 - ▶ Em pré-ordem
 - ▶ Em ordem
 - ▶ Em pós-ordem



Inserção

- ▶ A inserção começa na raiz.
 - ▶ Se a raiz é nula, o novo nó passa a ser a raiz da árvore.
 - ▶ Senão, encontra-se a **subárvore vazia** apropriada para inserção do novo nó
 - ▶ **Todo novo nó é uma folha!**
- ▶ **Exemplo:**
 - ▶ Desenhe a árvore binária de pesquisa que resulta a inserção sucessiva das chaves {90, 10, 170, 50, 80, 130, 100, 20, 30, 40, 70, 60 } em uma árvore inicialmente vazia.





Exercício

▶ **Exercício:**

- ▶ Desenhe a árvore binária de pesquisa que resulta a inserção sucessiva das chaves {Q U E S T A O F A C I L} em uma árvore inicialmente vazia.
- ▶ Mostre os percorrimientos em pré-ordem, ordem e pós-ordem

▶ **LISTA 17**





- ▶ <https://visualgo.net/bn/bst>

