



# Algoritmos e Estruturas de Dados I

Prof<sup>a</sup> Priscilla Abreu  
priscilla.abreu@ime.uerj.br  
2022.1

# Algoritmos e Estruturas de Dados I

---



## Roteiro da aula

- Árvores
  - Genéricas
  - Binárias



# LISTAS

---

# Algoritmos e Estruturas de Dados I

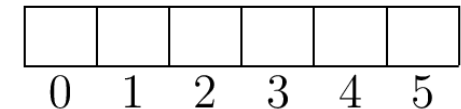


## Listas Linear

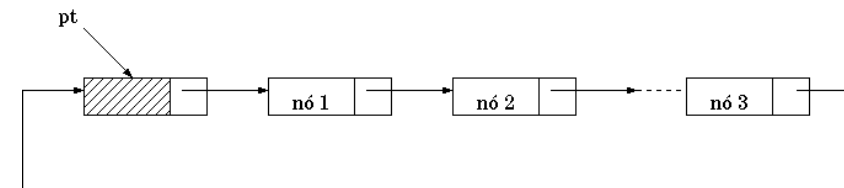
Estrutura que permite representar um conjunto de dados de forma a preservar a relação de ordem existente entre eles.

Uma lista é um exemplo de uma estrutura de dados linear, pois cada elemento tem:

- um predecessor único, exceto o primeiro elemento da lista;
- um sucessor único, exceto o último elemento.



As pilhas e filas são outros exemplos.



# Algoritmos e Estruturas de Dados I



## Listas Linear

### Listas lineares

#### Listas lineares gerais

SEM restrição de inserção e remoção de elementos

#### Listas particulares

COM restrição de inserção e remoção de elementos

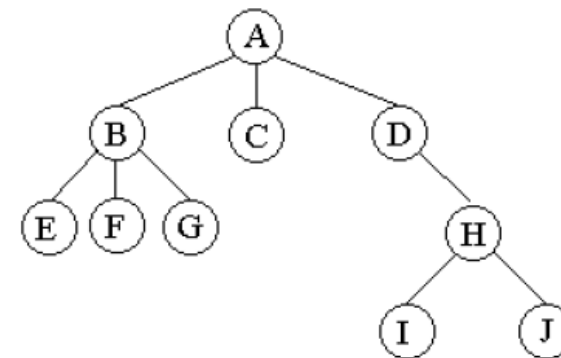
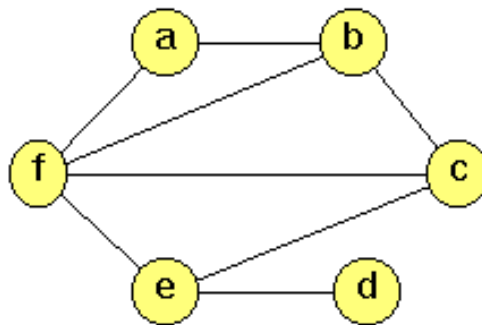
# Algoritmos e Estruturas de Dados I



## Listas Não Lineares

Em uma estrutura de dados não linear, os elementos, designados por nós, podem ter mais de um predecessor ou mais de um sucessor.

Grafos e árvores





# ÁRVORES

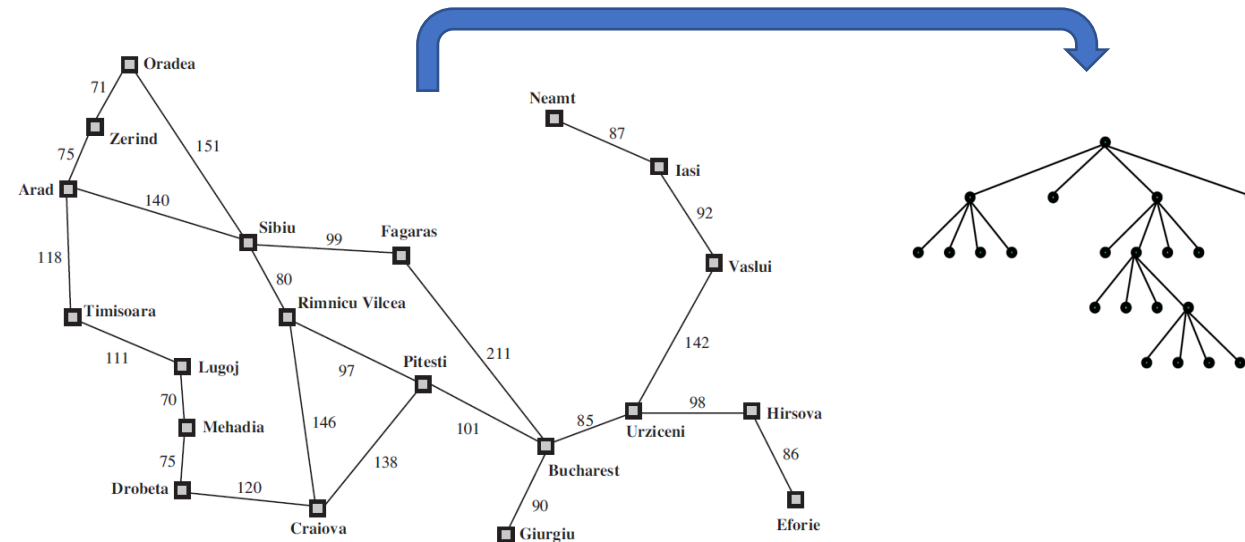
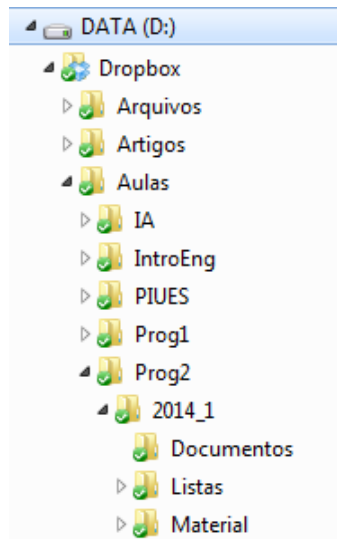
---

# Algoritmos e Estruturas de Dados I



## ÁRVORE

Um estrutura de dados do tipo árvore permite que dados sejam organizados de forma hierárquica.



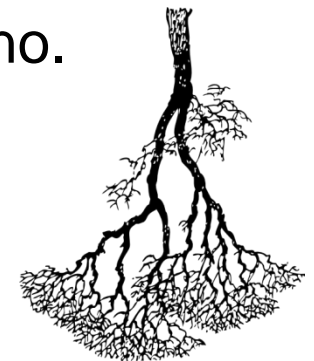
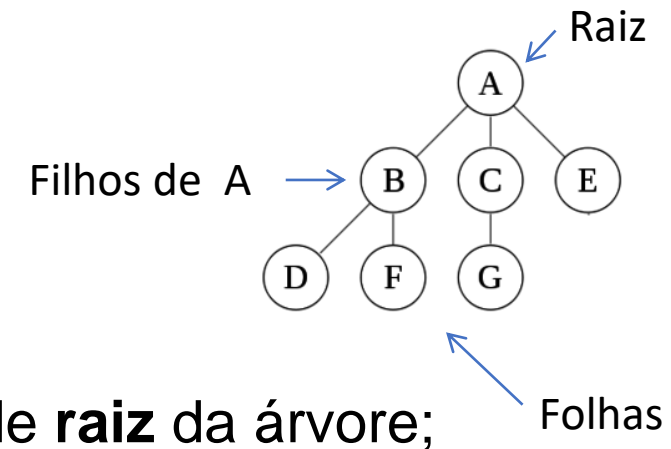


# Algoritmos e Estruturas de Dados I



## Árvores – conceitos

- Cada elemento de uma árvore é denominado **nó**;
- Toda árvore tem um elemento inicial que chamamos de **raiz** da árvore;
- Cada elemento da árvore pode ou não possuir nós abaixo dele hierarquicamente, denominados **filhos**.
- Os nós que não possuem filhos são denominados **folha** ou nó externo.
- Grau de um nó: número de filhos que ele possui.
- Grau da árvore: definido pelo nó de maior grau da árvore.

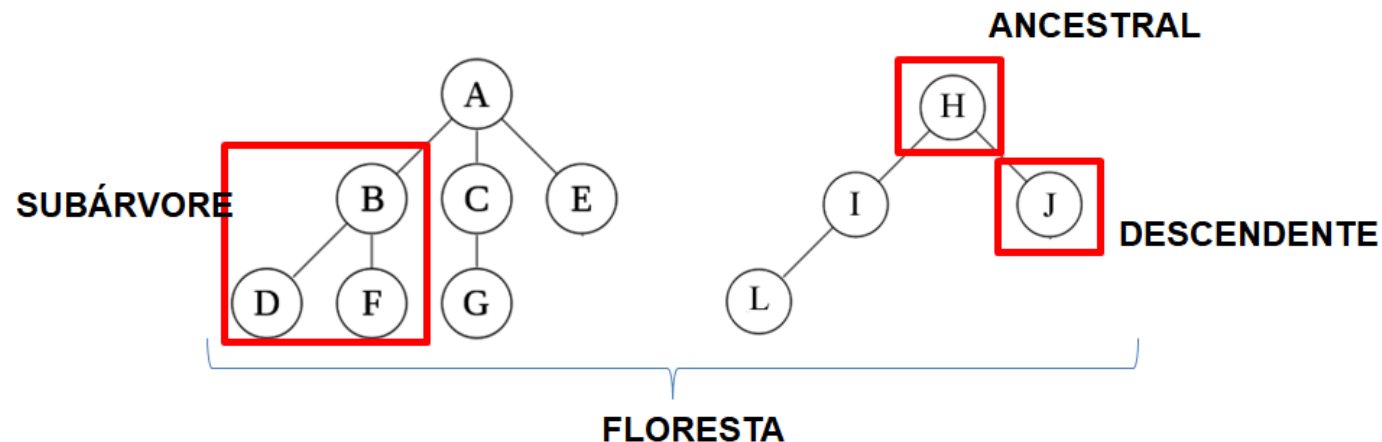


# Algoritmos e Estruturas de Dados I



## Árvores – conceitos

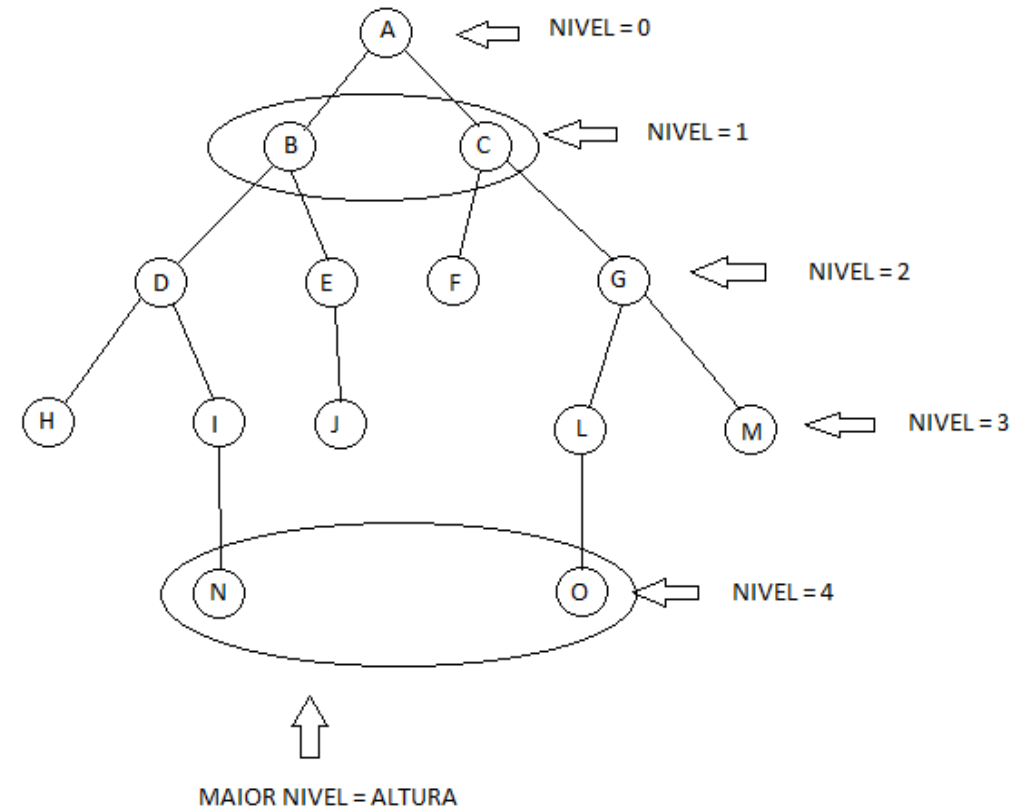
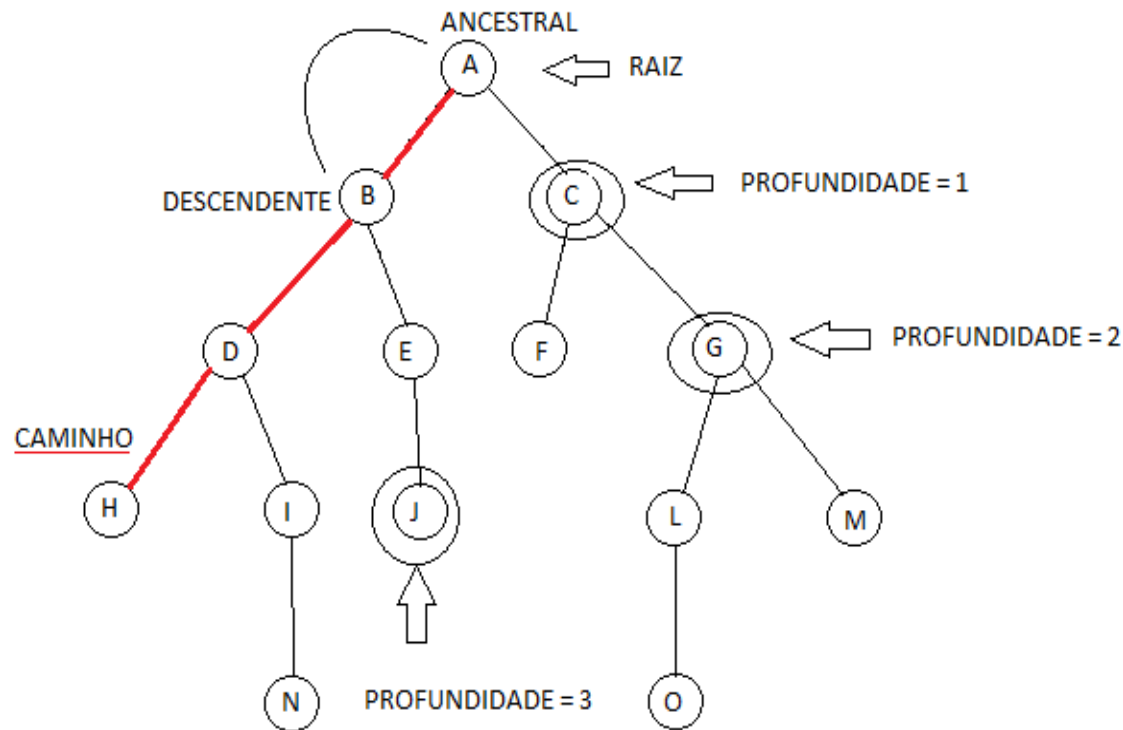
- Subárvore: conjunto de nós formado a partir de um determinado nó.
- Ancestral: nó que antecede um determinado nó.
- Descendente: nó que sucede um determinado nó.
- Floresta: o conjunto de árvores disjuntas.



# Algoritmos e Estruturas de Dados I

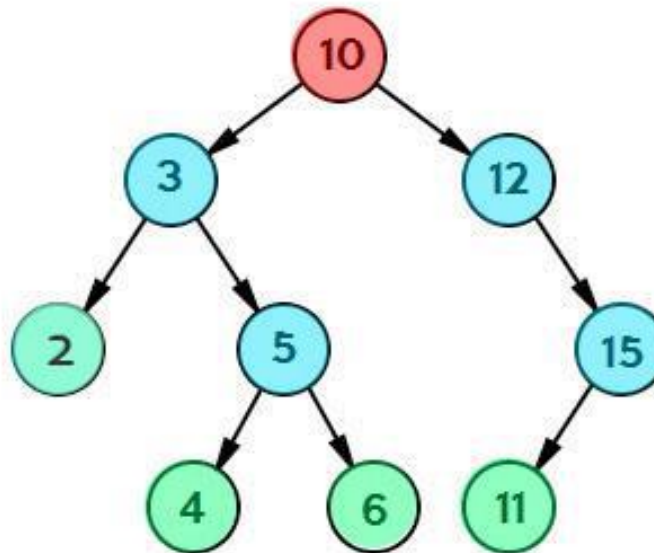


## Árvores – conceitos



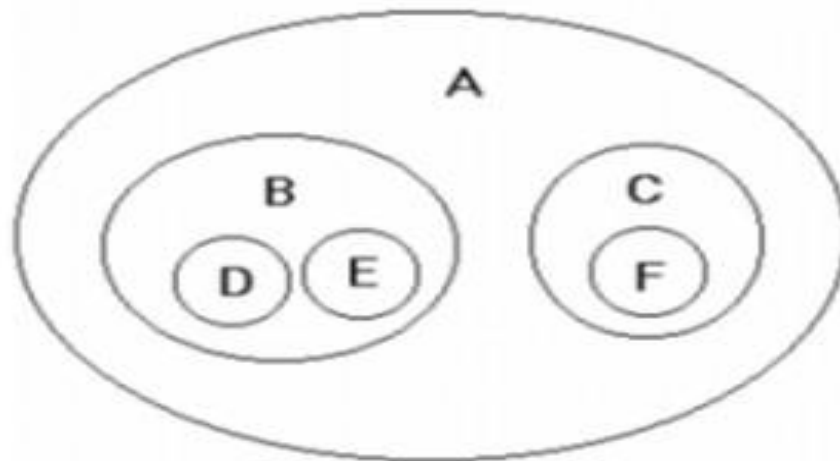
## Árvores – representação

- Hierarquia



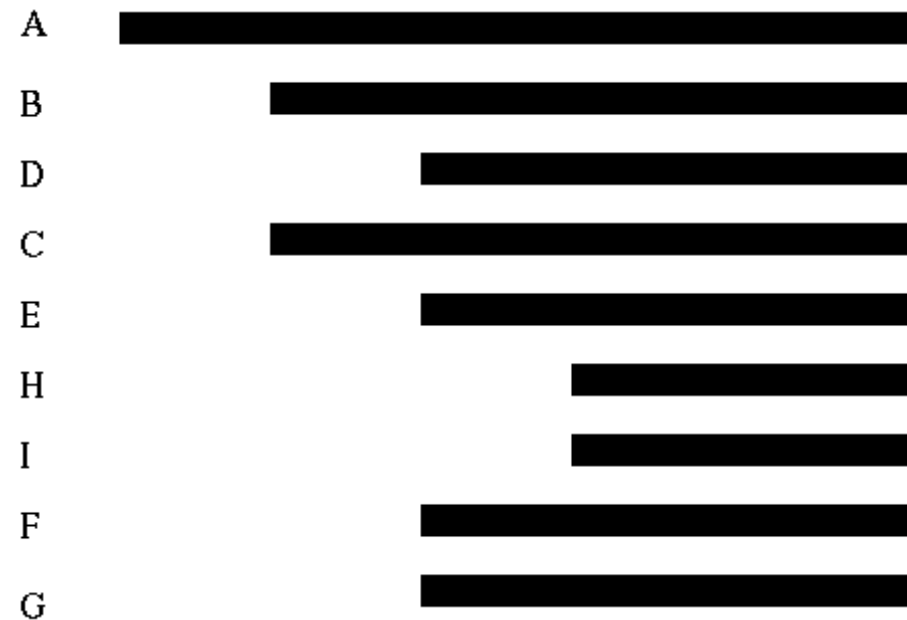
## Árvores – representação

- Diagrama de Venn



## Árvores – representação

- Gráfico de Barras



# Algoritmos e Estruturas de Dados I



## Árvores – representação

- Expressão com parênteses

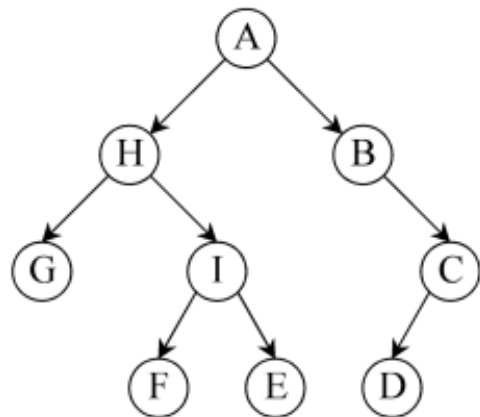
**( A ( B ( D ( ) E ( ) ) ) ( C ( F ( ) ) ) ) )**

# Algoritmos e Estruturas de Dados I



## Árvores Binárias

Estrutura de dados que é constituída por um conjunto finito de nós, em que cada nó pode ter no máximo **dois** filhos, ou sub-árvores: a sub-árvore da **direita** (sad) e a sub-árvore da **esquerda** (sae).

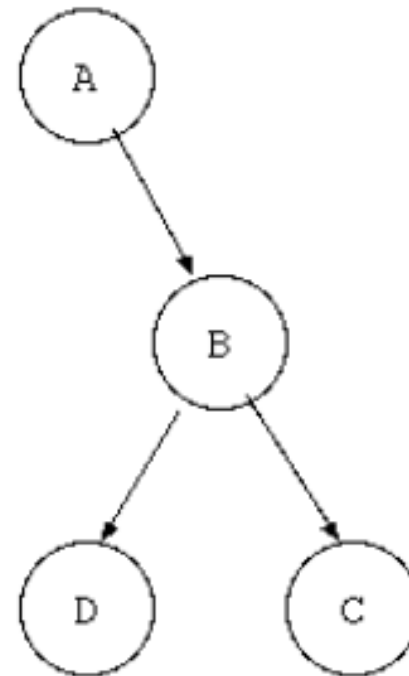
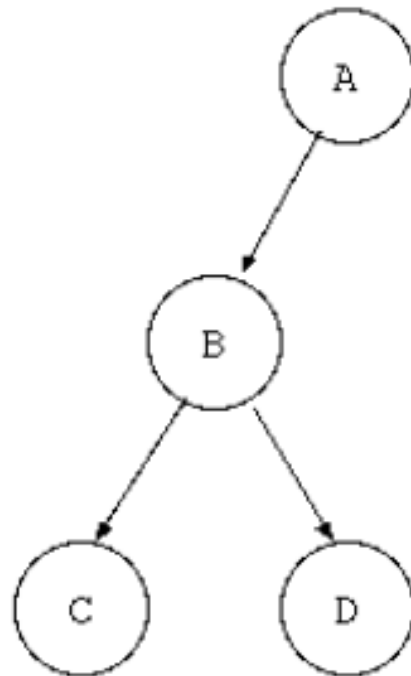




# Algoritmos e Estruturas de Dados I

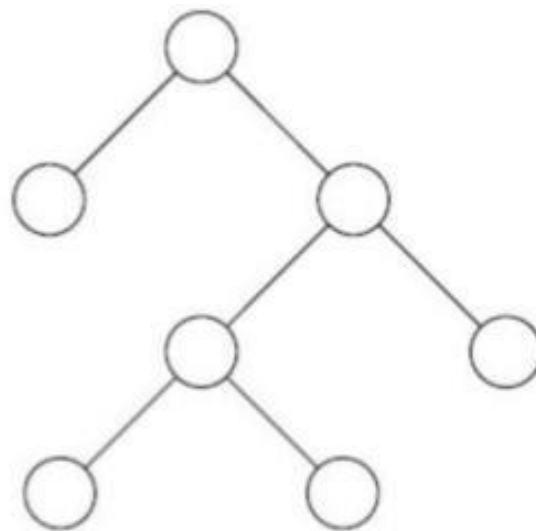


## Árvore X Árvore Binária



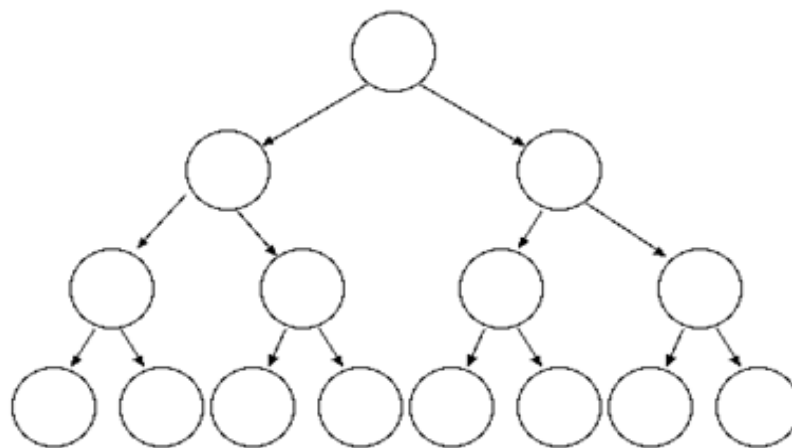
## Árvore Estritamente Binária

Árvore binária em que cada nó tem 0 ou 2 filhos.



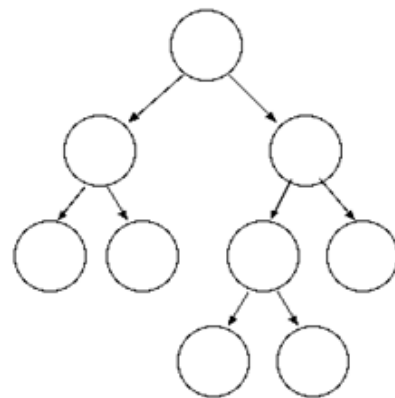
## Árvore Binária Cheia

Árvore estritamente binária em que se um nó tem alguma subárvore vazia então ele está no último nível.



## Árvore Binária Completa

Árvore binária completa: árvore em que se  $n$  é um nó com algumas de suas subárvores vazias, então  $n$  se localiza no penúltimo ou no último nível. Portanto, toda árvore cheia é completa e estritamente binária.



# Algoritmos e Estruturas de Dados I

---



## Árvore Binária – Percurso

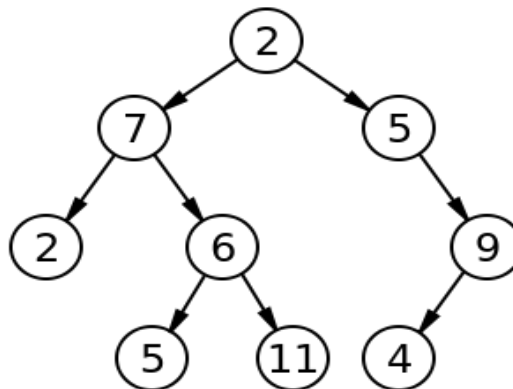
- PRÉ ORDEM
- EM ORDEM
- PÓS ORDEM

## Árvore Binária – Percurso

- PRÉ ORDEM

No percurso em pré-ordem, primeiramente a raiz é visitada; depois, a sub-árvore esquerda; e finalmente, a sub-árvore direita.

No exemplo, o percurso seria feito na seguinte ordem: 2, 7, 2, 6, 5, 11, 5, 9 e 4.

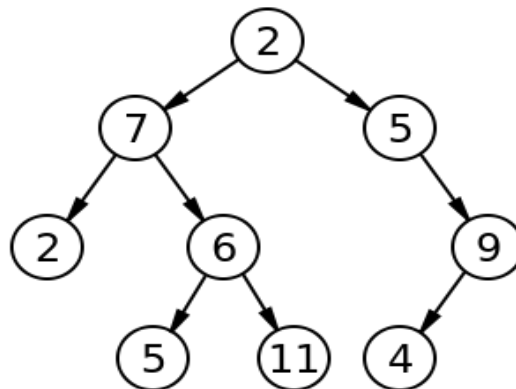


## Árvore Binária – Percurso

- EM ORDEM (SIMÉTRICO)

No percurso simétrico (em ordem), primeiro é visitada a sub-árvore esquerda; logo após, a raiz; por final, a sub-árvore direita.

No exemplo, o percurso seria feito na seguinte ordem: 2, 7, 5, 6, 11, 2, 5, 4 e 9.

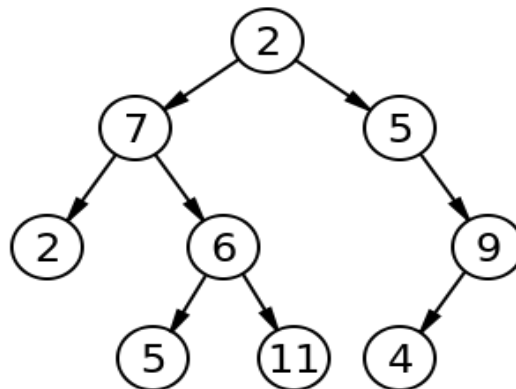


## Árvore Binária – Percurso

- PÓS ORDEM

O percurso em pós-ordem inicia-se visitando a sub-árvore esquerda; em seguida, a sub-árvore direita; encerrando, a raiz é visitada.

No exemplo, o percurso seria feito na seguinte ordem: 2, 5, 11, 6, 7, 4, 9, 5 e 2.

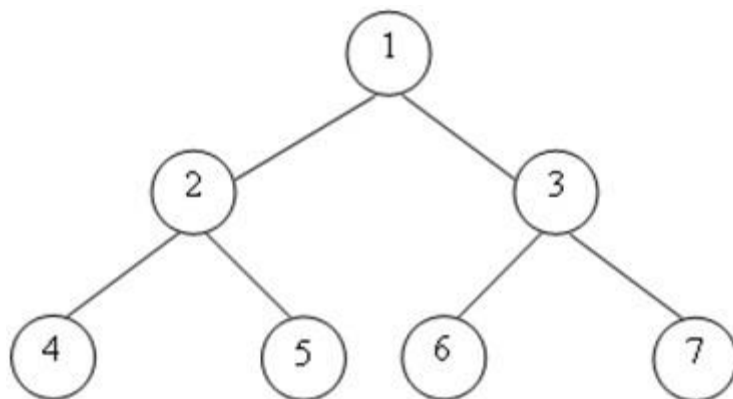




## Árvore Binária – Percurso

- EXERCÍCIO

Qual a ordem do percurso da árvore abaixo se utilizarmos o percurso simétrico?





# ÁRVORE BINÁRIA DE BUSCA

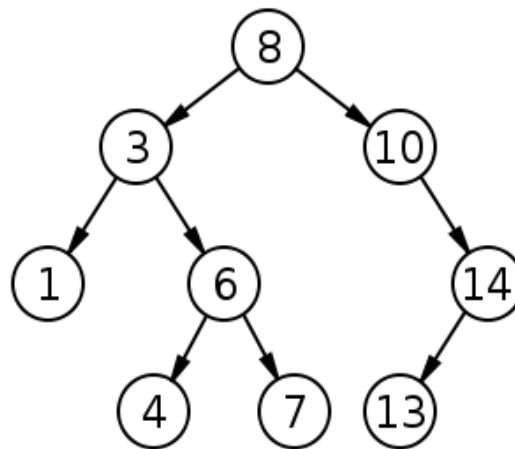
---

# Algoritmos e Estruturas de Dados I



## Árvore Binária de Busca

Árvore binária baseada em nós, onde todos os nós da subárvore esquerda possuem um valor numérico inferior ao nó raiz e todos os nós da subárvore direita possuem um valor superior ao nó raiz.

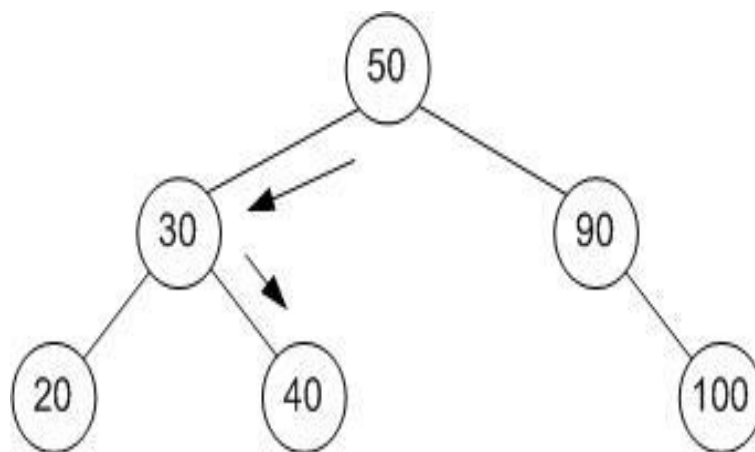


# Algoritmos e Estruturas de Dados I



## Árvore Binária de Busca

Busca



# Algoritmos e Estruturas de Dados I



## Árvore Binária de Busca

### Inserção

Considere que iremos inserir os seguintes elementos nessa ordem:

50 60 30 40 70 20

# Algoritmos e Estruturas de Dados I



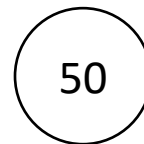
## Árvore Binária de Busca

### Inserção

Considere que iremos inserir os seguintes elementos nessa ordem:

50 60 30 40 70 20

Nó Raiz



# Algoritmos e Estruturas de Dados I



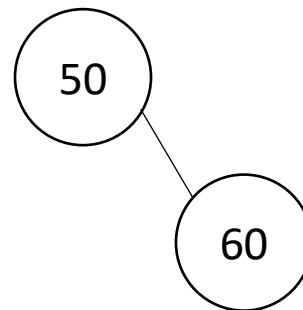
## Árvore Binária de Busca

### Inserção

Considere que iremos inserir os seguintes elementos nessa ordem:

50 60 30 40 70 20

Nó Raiz



$60 > 50 = \text{SIM}$

A direita

# Algoritmos e Estruturas de Dados I

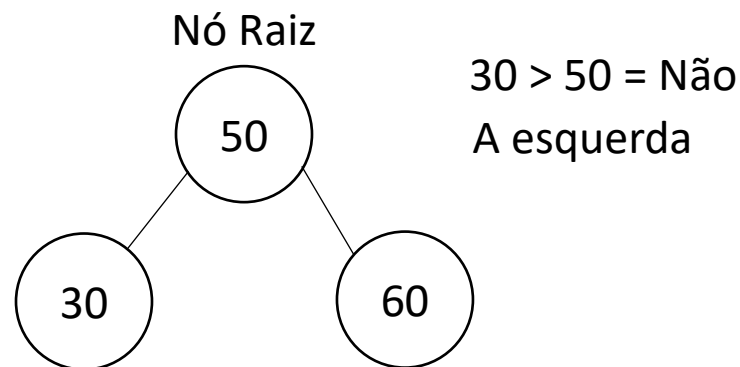


## Árvore Binária de Busca

### Inserção

Considere que iremos inserir os seguintes elementos nessa ordem:

50 60 30 40 70 20





# Algoritmos e Estruturas de Dados I

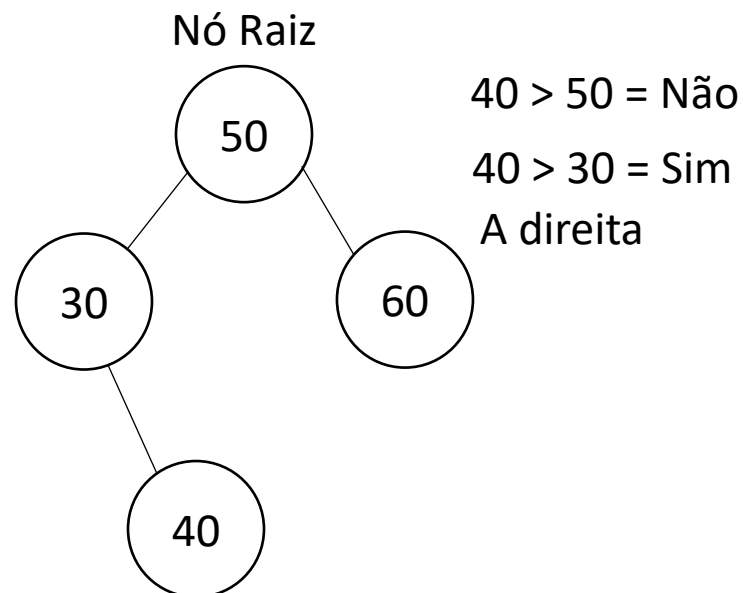


## Árvore Binária de Busca

### Inserção

Considere que iremos inserir os seguintes elementos nessa ordem:

50 60 30 **40** 70 20



# Algoritmos e Estruturas de Dados I

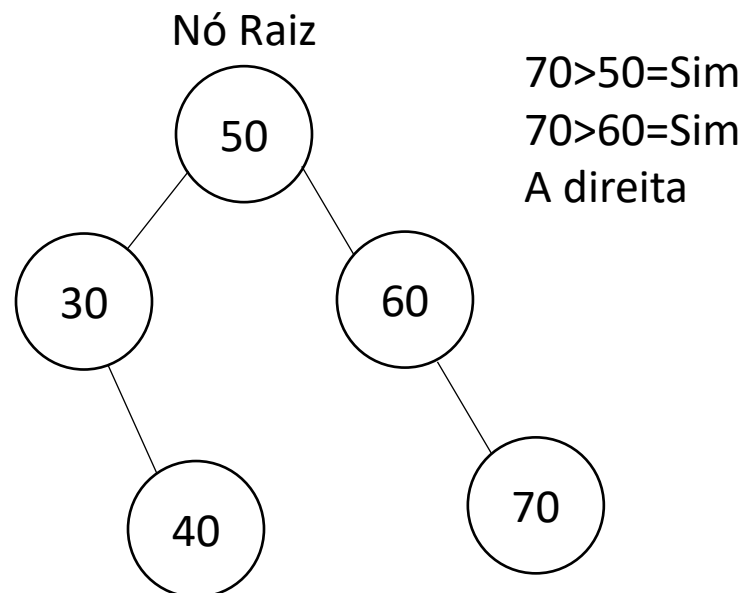


## Árvore Binária de Busca

### Inserção

Considere que iremos inserir os seguintes elementos nessa ordem:

50 60 30 40 **70** 20



# Algoritmos e Estruturas de Dados I

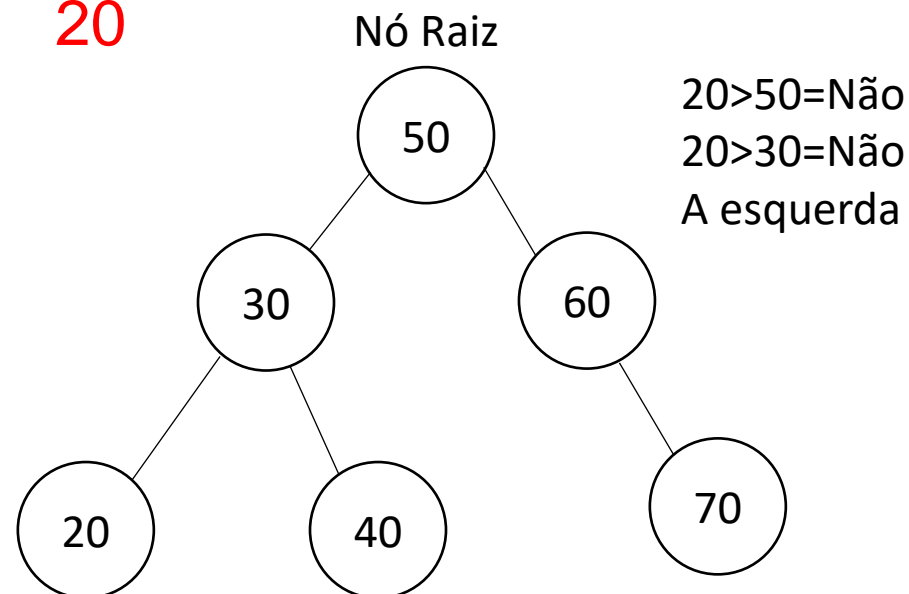


## Árvore Binária de Busca

### Inserção

Considere que iremos inserir os seguintes elementos nessa ordem:

50 60 30 40 70 **20**





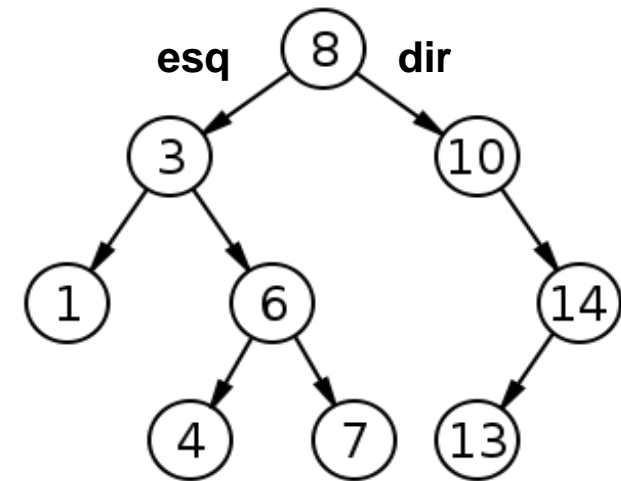
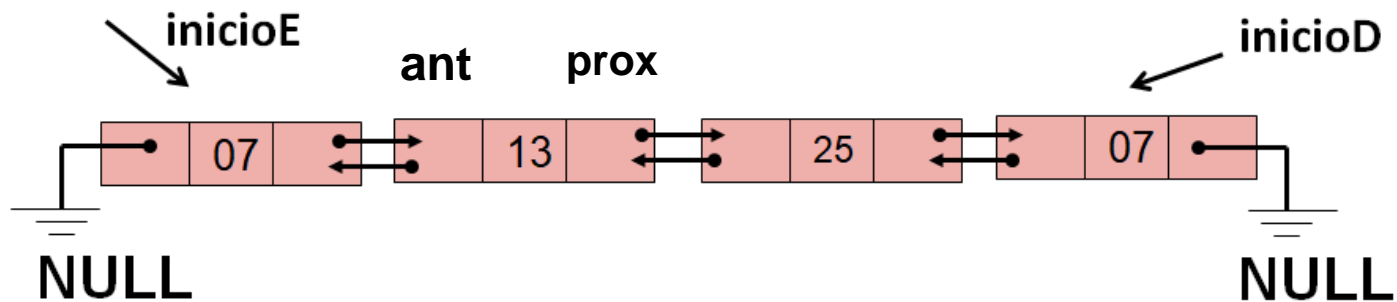
Como podemos definir uma *struct* para representar uma árvore binária?

---

# Algoritmos e Estruturas de Dados I



Alguma semelhança???



# Algoritmos e Estruturas de Dados I



---

```
#include <stdio.h>
#include <stdlib.h>
typedef struct no{
    int info;
    struct no *esq, *dir;
}no;
no *raiz;
int main(){
}
```



**Mais informações na  
próxima aula...**

---

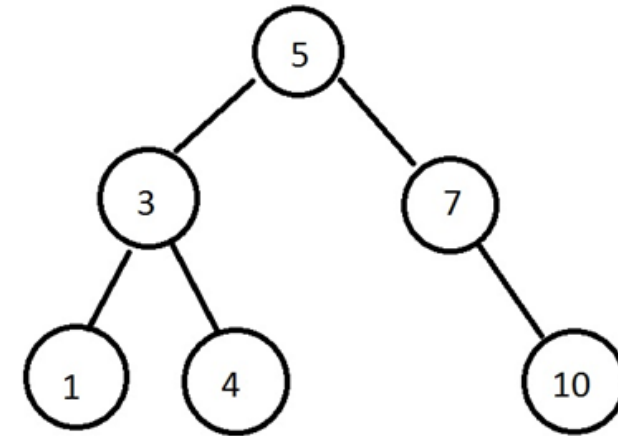
# Algoritmos e Estruturas de Dados I



## EXERCÍCIOS

Considere a árvore ao lado:

- Quantos nós folha a árvore têm?
- Qual o grau da árvore?
- Qual sua altura?
- Dê um exemplo de uma subárvore
- dessa árvore.





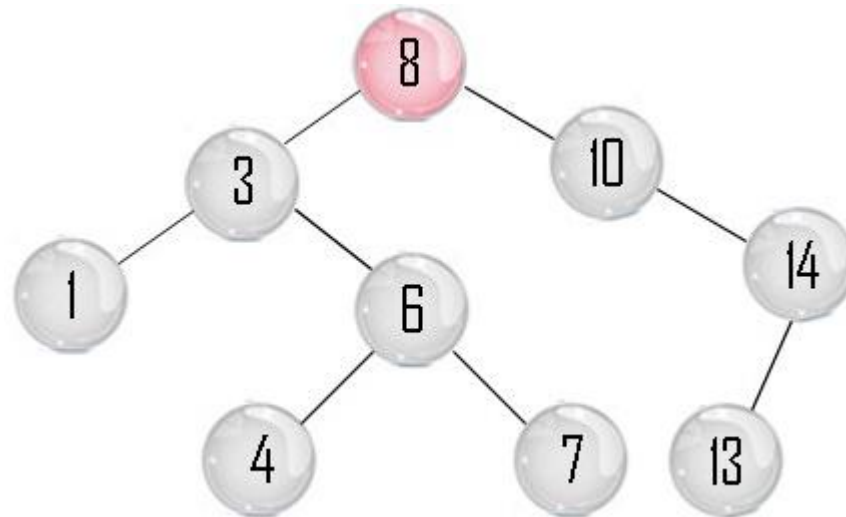
# Algoritmos e Estruturas de Dados I



## EXERCÍCIOS

Para a árvore a seguir, você deve apresentar como ocorrerá cada um dos percursos abaixo.

- Pré-ordem
- Em ordem (Simétrico)
- Pós-ordem



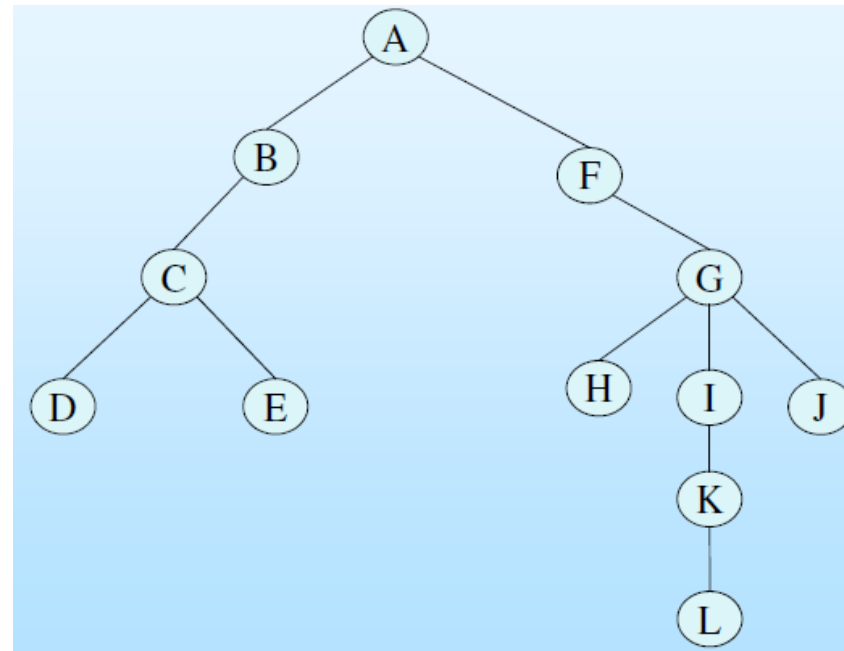
# Algoritmos e Estruturas de Dados I



## EXERCÍCIOS

Determine:

- a) A altura da árvore.
- b) Altura do nó G.
- c) Nível do nó G.
- d) Nível do nó A.
- e) Altura do nó E.
- f) Mostre as subárvores do nó F.
- g) Essa é uma árvore binária de busca? Justifique sua resposta.





# DÚVIDAS???

---



# FIM