Fatec
Araras

# Estrutura de Dados

Prof. Orlando Saraiva Júnior
orlando.saraiva@unesp.br

""A program is like a poem:

you cannot write a poem

without writing it."

E. W. Dijkstra

# Estrutura de Dados

# Objetivo da aula

Fatec Araras

Conhecer Árvores

- Árvores
- Árvores binárias

Árvores Binárias em C++

Árvores Binárias em Python

# Árvores

Fatec Araras

Uma **árvore** é uma coleção finita de n>=0 nodos. Se n=0, dizemos que a árvore é nula. Caso contrário, a árvore apresenta as seguintes características:

→ existe um nodo especial chamado raiz;

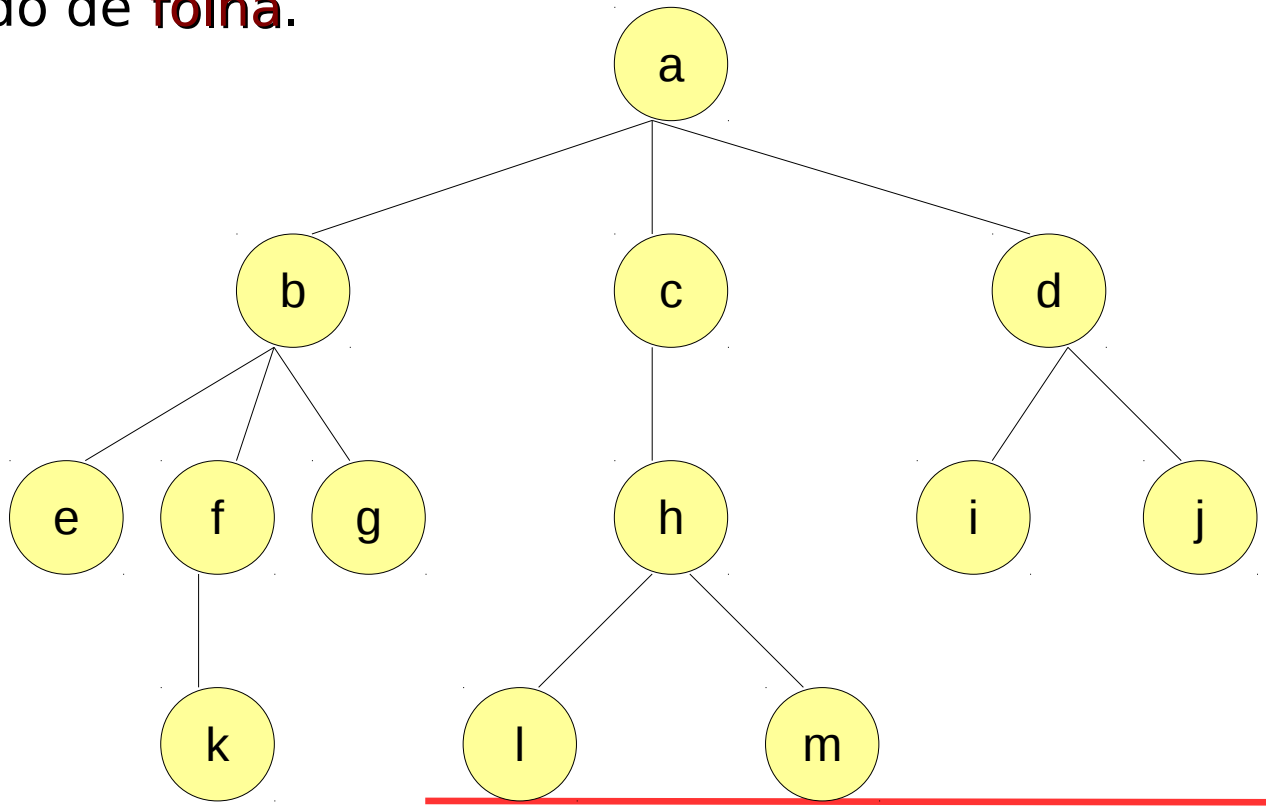
→ os demais são particionais em $T_1$,$T_2$,... , $T_k$ estruturas disjuntas de árvores;

→ As estruturas $T_1$,$T_2$,... , $T_k$ denominam-se **subárvores**.

A exigência de que as estruturas $T_1$,$T_2$,... , $T_k$ sejam coleções disjuntas, garante

# Árvores

O número de subárvores de um nodo denomina-se **grau.** Um nodo que possui grau 0, ou seja, não possui sub-árvores é chamado de **folha**.



# Árvores

Na figura anterior:

•O nó a possui grau 3.

•O nó d possui grau 2.

•São folhas os nós e, k, g, l, m, i e j.

As raízes das subárvores de um nodo denominam-se **filhos** do nó, que é o **pai** delas.

•Os filhos do nó b são: e,f e g.

•O nó h é pai de l e m.

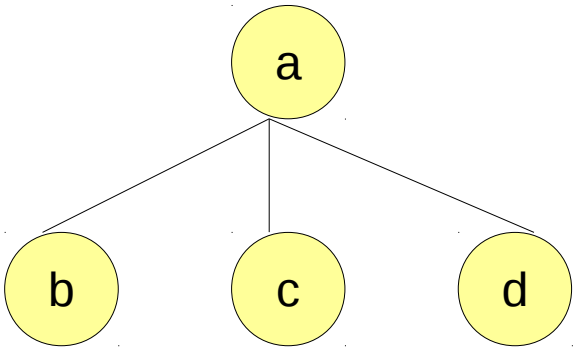
Por definição, dizemos que a raiz de uma árvore encontra-se no nível 1. A altura de uma árvore é definida como sendo máximo de níveis. Na figura anterior, a altura da árvore é 4.

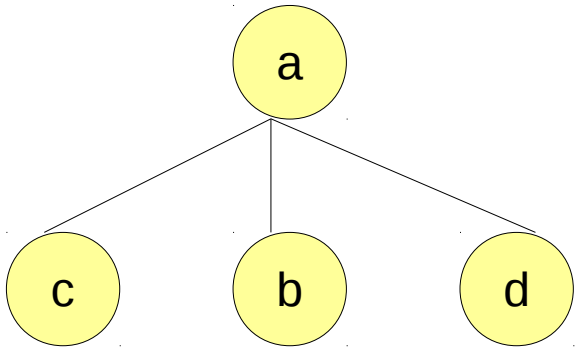
# Árvores Ordenadas

Fatec Araras

Uma árvore é ordenada quando a ordem das subárvores é importante.

Por exemplo, as duas árvores ordenadas seguintes são diferentes:



# Árvore Binária

Fatec Araras

Uma **árvore binária** é uma árvore que pode ser nula, ou então tem as seguintes características:

Existe um nodo especial, chamado raiz;

Os demais nodos são particionais em $T_1$, $T_2$ estruturas disjuntas de árvores binárias.

$T_1$ é denominada **árvore esquerda** e $T_2$, **árvore direita** da raiz.

A árvore binária é um caso especial de árvore em que nenhum nodo tem grau superior a 2, isto é, nenhum nó tem mais de dois filhos.

Logo, árvore binária é uma árvore ordenada de grau 2.

# Formas de percorrer

Em algumas aplicações, é necessário percorrer uma árvore de forma sistemática, visitando cada nó da árvore uma única vez, em determinada ordem.

Veremos as seguintes formas:

- Pré-ordem.
- In-ordem ou ordem simétrica.
- Pós-ordem.

# Pré-ordem

Para percorrer uma árvore binária em pré-ordem:

1) Viste a raiz.

2) Percorra a sua subárvore esquerda em pré-ordem.

3) Percorra a sua subárvore direita em pré-ordem.

Visitar um nó significa executar uma certa ação no nó.

# In-ordem

Fatec Araras

Percorrer uma árvore binária em in-ordem:

1) Percorrer a sua subárvore esquerda em in-ordem.

2) Vistar a raiz.

3) Percorrer a sua subárvore direita em in-ordem.

A in-ordem visita a raiz entre as ações de percorrer as duas subárvores. É conhecida também pelo nome de ordem simétrica.

# Pós-ordem

Fatec Araras

Percorrer uma árvore binária em pós-ordem:

1) Percorrer a sua subárvore esquerda em pós-ordem.

2) Percorrer a sua subárvore direita em pós-ordem.

3) Vistar a raiz.

# Hora do código

# Figurinhas

Prof. Me. Orlando Saraiva Júnior

# Árvores em C++

Fatec Araras

Código: arvore_orlando.cpp

```
node * inicializar(int valor)
void destruir_arvore(node *leaf)
void inserir(int key, node *leaf)
node * search(int key, node *leaf)
void ordem(node *leaf)
void pos_ordem(node *leaf)
void pre_ordem(node *leaf)
node* encontrar_menor(node* leaf)
node* remover(int x, node* leaf)
```

Fatec
Araras

# Implementando árvores com Python

Prof. Me. Orlando Saraiva Júnior

Fatec
Araras

# Dúvidas

**Prof. Orlando Saraiva Júnior**
orlando.saraiva@unesp.br