

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

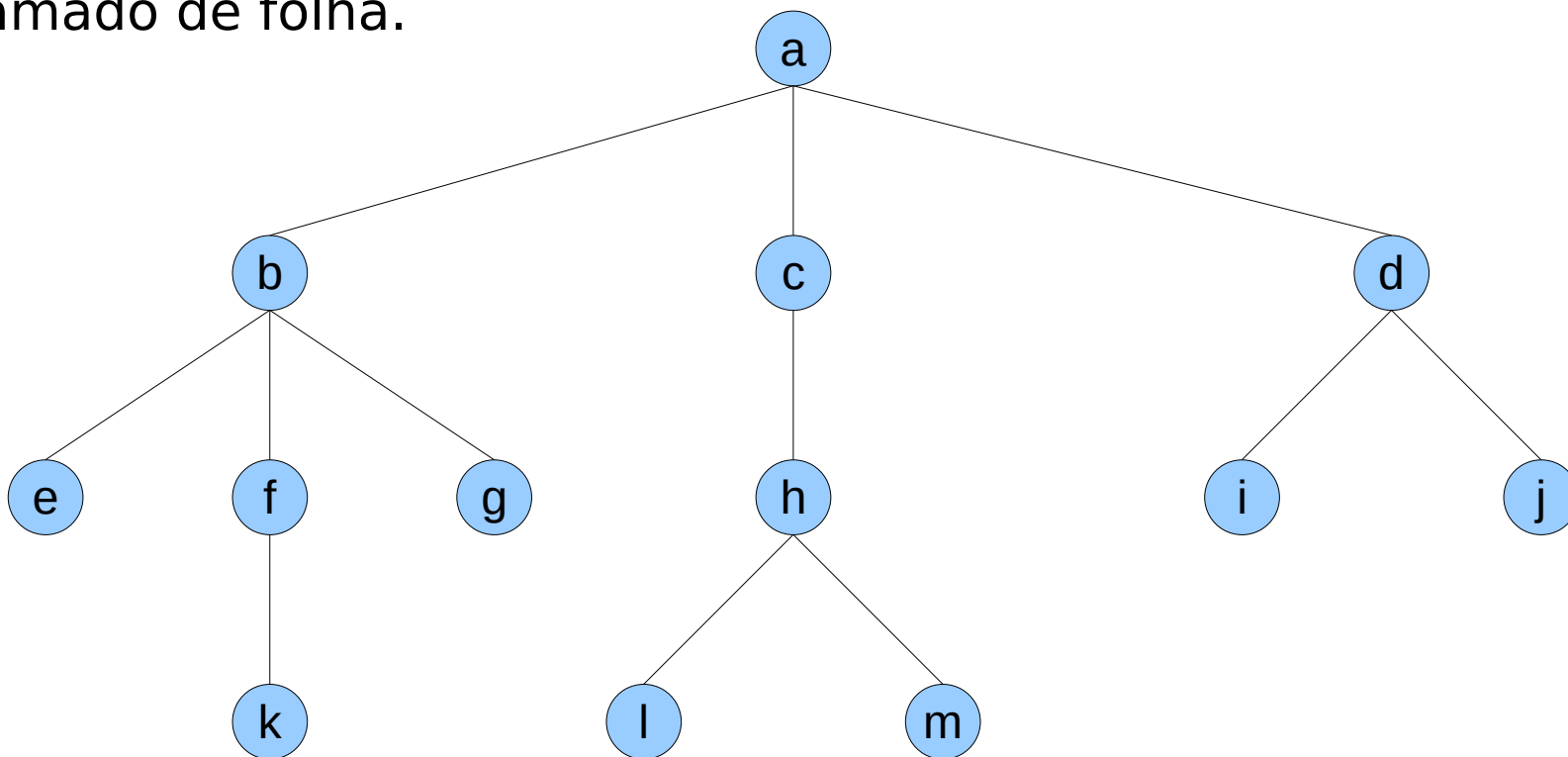
Conhecer Árvores

- Árvores
- Árvores binárias

Uma **árvore** é uma coleção finita de $n \geq 0$ 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, \dots, T_k estruturas disjuntas de árvores;
- As estruturas T_1, T_2, \dots, T_k denominam-se subárvores .

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.



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.

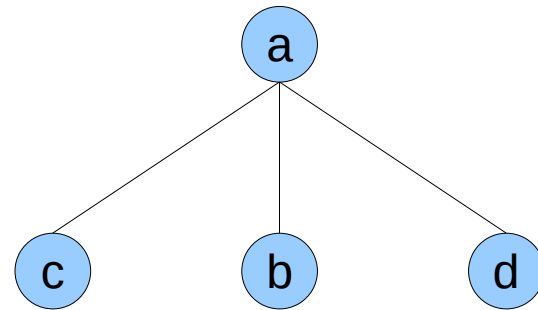
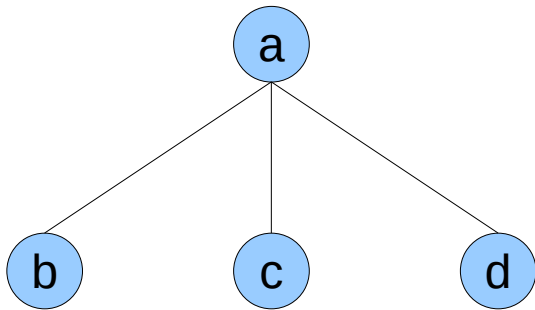
Os filhos do nó b são: e, f e g.

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

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



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**.

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.

Para percorrer uma árvore binária em pré-ordem:

- 1) Visite 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.

Percorrer uma árvore binária em in-ordem:

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

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.

Dúvidas

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

1) Ajuste o programa `arvore_binaria.cpp`, para a seguinte estrutura:

```
struct Node {  
    string nome;  
    int idade;  
    Node* left;  
    Node* right;  
};
```

2) Adapte a função **search** para realizar a busca por idade e crie a função **search2** para realizar a busca por nome.

3) Adaptar o código para que percorra in-ordem, pré-ordem e pós-ordem pelo campo idade.