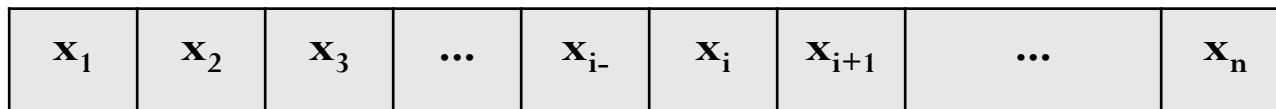


# Introdução a Árvore

# Introdução

- Lista lineares: organização linear dos dados, onde sua propriedade básica é a relação sequencial mantida entre seus elementos
  - Um elemento da lista está após o outro
  - Não há relação hierárquica

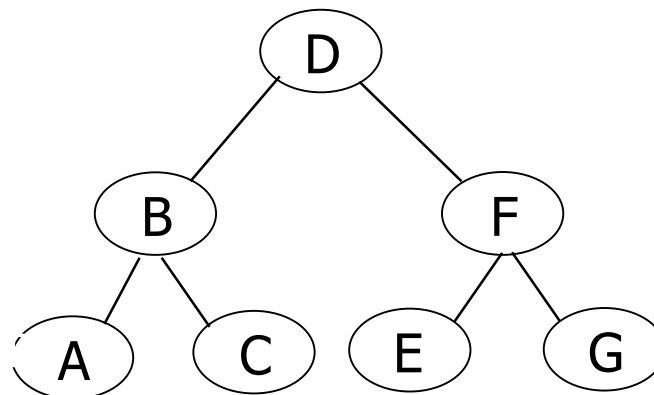


# Introdução

- Estruturas lineares:
  - Listas
  - Pilhas
  - Filas
  - Listas encadeadas circulares
  - Listas duplamente encadeadas

# Introdução

- Listas não-lineares
  - Estrutura em grafos
  - **Estrutura em árvores**
- **Estrutura em árvore:** organização dos dados de forma não-linear, mantendo um relacionamento hierárquico entre os elementos



# Introdução

- Exemplos de estrutura em árvore
  - Árvore genealógica
  - Organograma de uma empresa

# Introdução

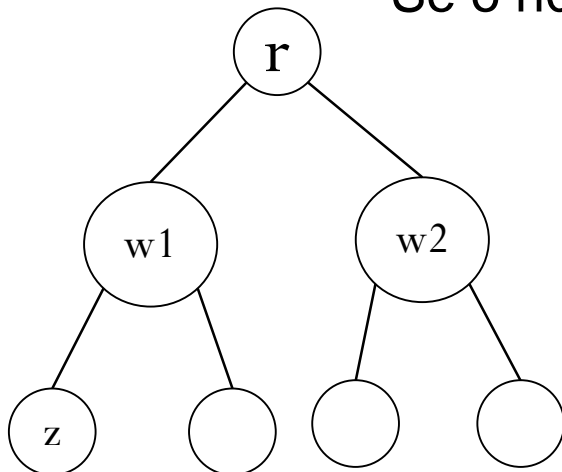
- Motivação/Vantagens:
  - Representatividade no relacionamento entre os dados
  - Favorece a extração de informação de forma eficiente
    - Quem são os filhos de Maria?
    - Onde está o capítulo de estrutura em árvores?
    - Quem é o diretor da seção de financeiro?

# Definição

- Uma árvore enraizada é um conjunto finito de elementos denominados nós ou vértices tais que:
  - $T = \emptyset$ , a árvore é dita vazia, ou
  - $T = \{r\} \cup \{T_1\} \cup \{T_2\} \cup \{T_3\} \cup \dots \cup \{T_n\}$ ,  $n > 0$ .
- A definição de árvore é recursiva
  - $r$  é a raiz da árvore
  - Cada conjunto  $T_1, T_2, T_3, \dots, T_n$  disjuntos são subárvores de  $r$
- Uma floresta é um conjunto de árvores

# Terminologias

- Nós filhos, pais, tios, irmãos e avôs
  - Seja  $r$  o nó raiz de uma subárvore  $T$ 
    - Os nós raízes  $w_1, w_2, \dots, w_j$  das subárvores de  $r$  são chamados filhos de  $r$
    - O nó  $r$  é chamado de pai de  $w_1, w_2, \dots, w_j$
    - Os nós  $w_1, w_2, \dots, w_j$  são ditos irmãos
    - Se o nó  $z$  é filho de  $w_1$ , então  $w_2$  é tio de  $z$  e  $r$  é avô de  $z$





# Terminologias

- Grau de saída: o número de filhos de um nó é chamado de grau desse nó
- Descendente: o grau de uma árvore é o máximo entre os graus de seus nós
- Ancestral: se o nó  $x$  pertence à uma subárvore do nó  $v$ , então  $x$  é descendente de  $v$  e  $v$  é ancestral

# Terminologias

- Nó folha: um nó que não possui descendentes é chamado de nó folha, ou seja, um nó folha é aquele com grau de saída nulo ou zero
- Nó interior: um nó que não é folha é chamado de nó interior, nó interno, ou ainda, nó intermediário

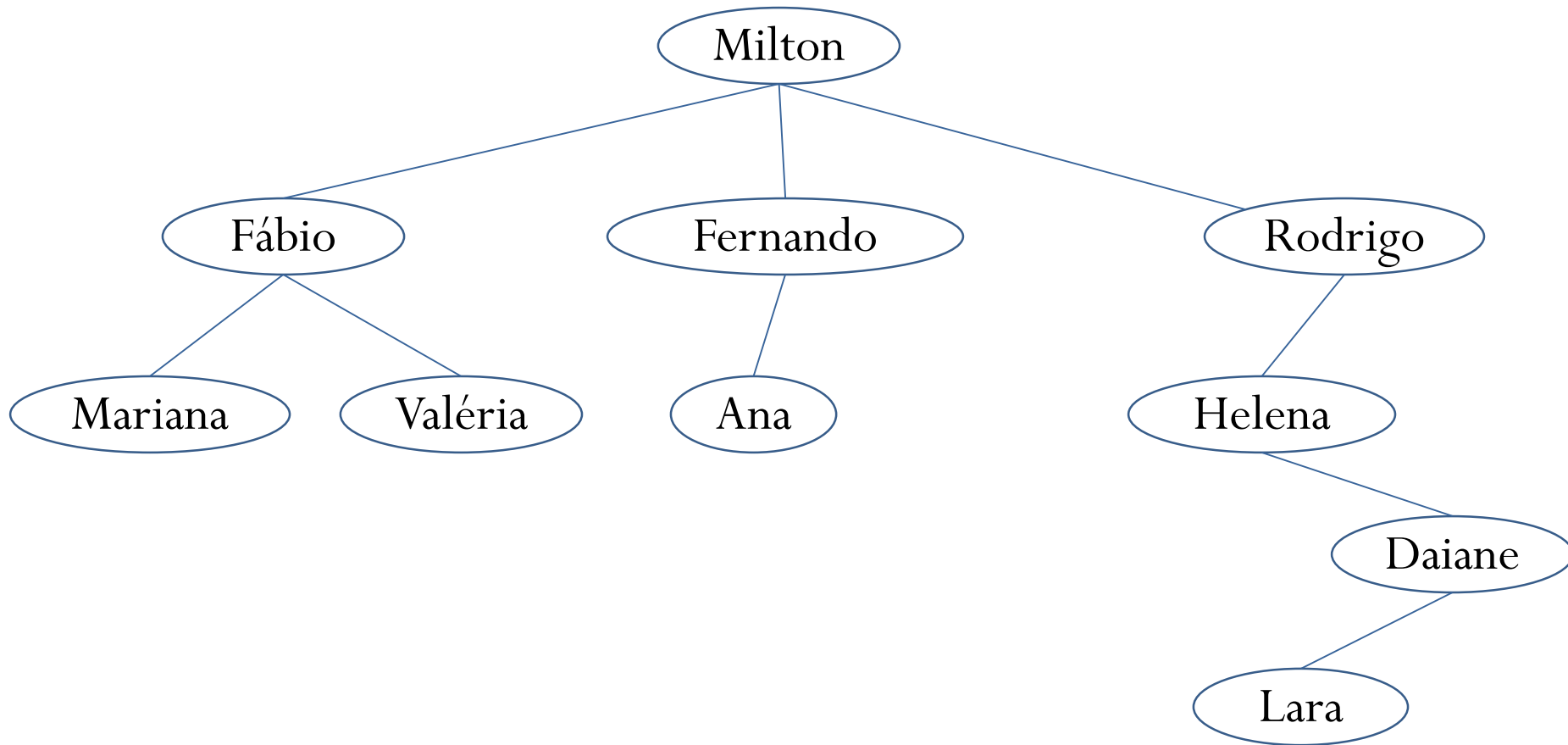
# Terminologias

- Caminho: uma sequência de nós distintos  $w_1, w_2, \dots, w_j$ , tal que existe sempre entre nós consecutivos a relação “é filho de” ou é “pai de”, é denominada um caminho na árvore: diz-se que  $w_1$  alcança  $w_j$  e que  $w_j$  é alcançado por  $w_1$ .
- Comprimento do caminho: um caminho de  $k$  vértices é obtido pela sequência de  $k-1$  pares; o valor  $k-1$  é o comprimento do caminho

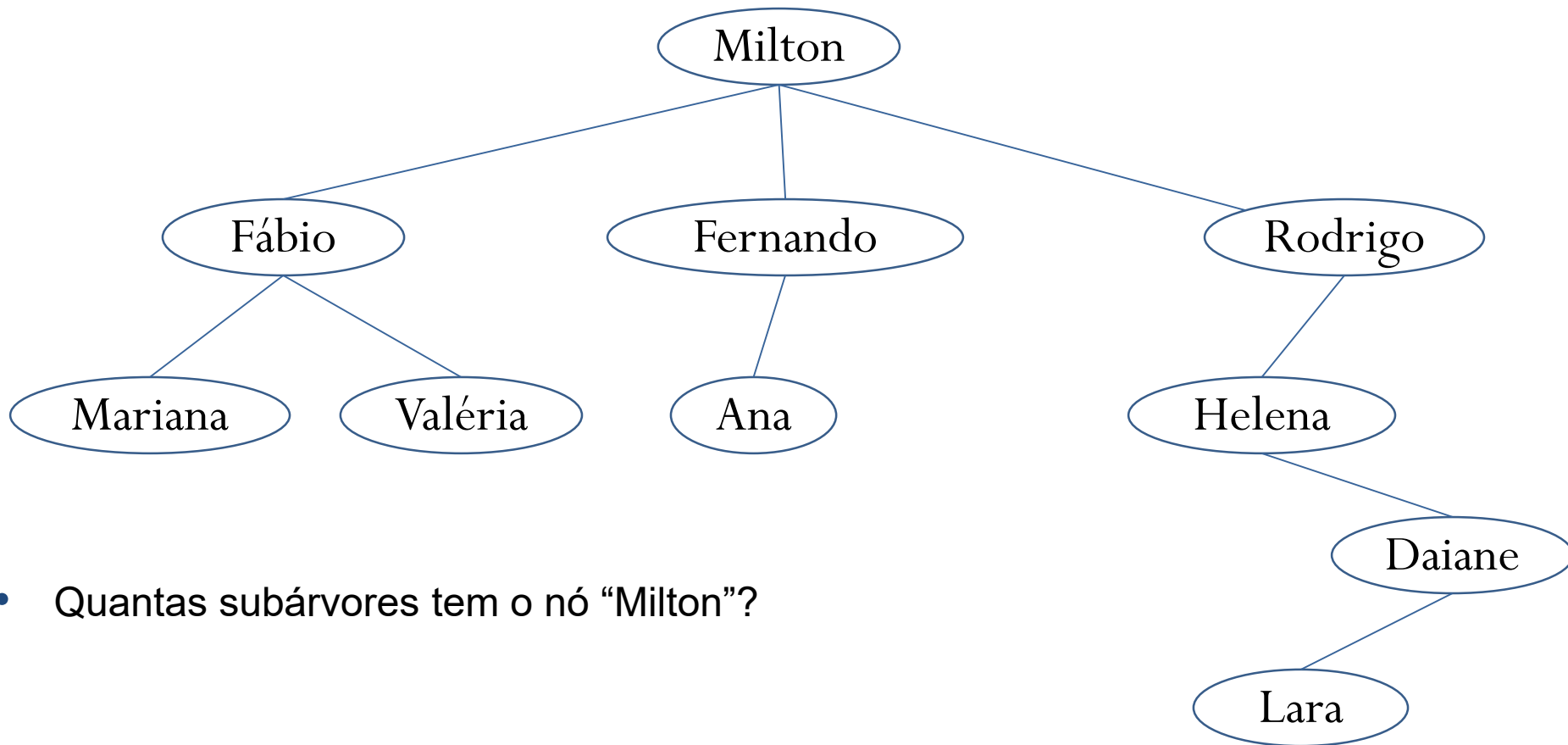
# Terminologias

- Nível (ou profundidade) e altura de um nó
  - O nível de um nó é o tamanho do caminho entre a raiz da árvore até esse nó
    - A raiz tem nível 0
  - A altura de um nó é o tamanho do maior caminho entre este nó e uma folha descendente desse nó
    - As folhas têm altura 0
    - A altura da raiz equivale a altura da árvore.

# Exemplo

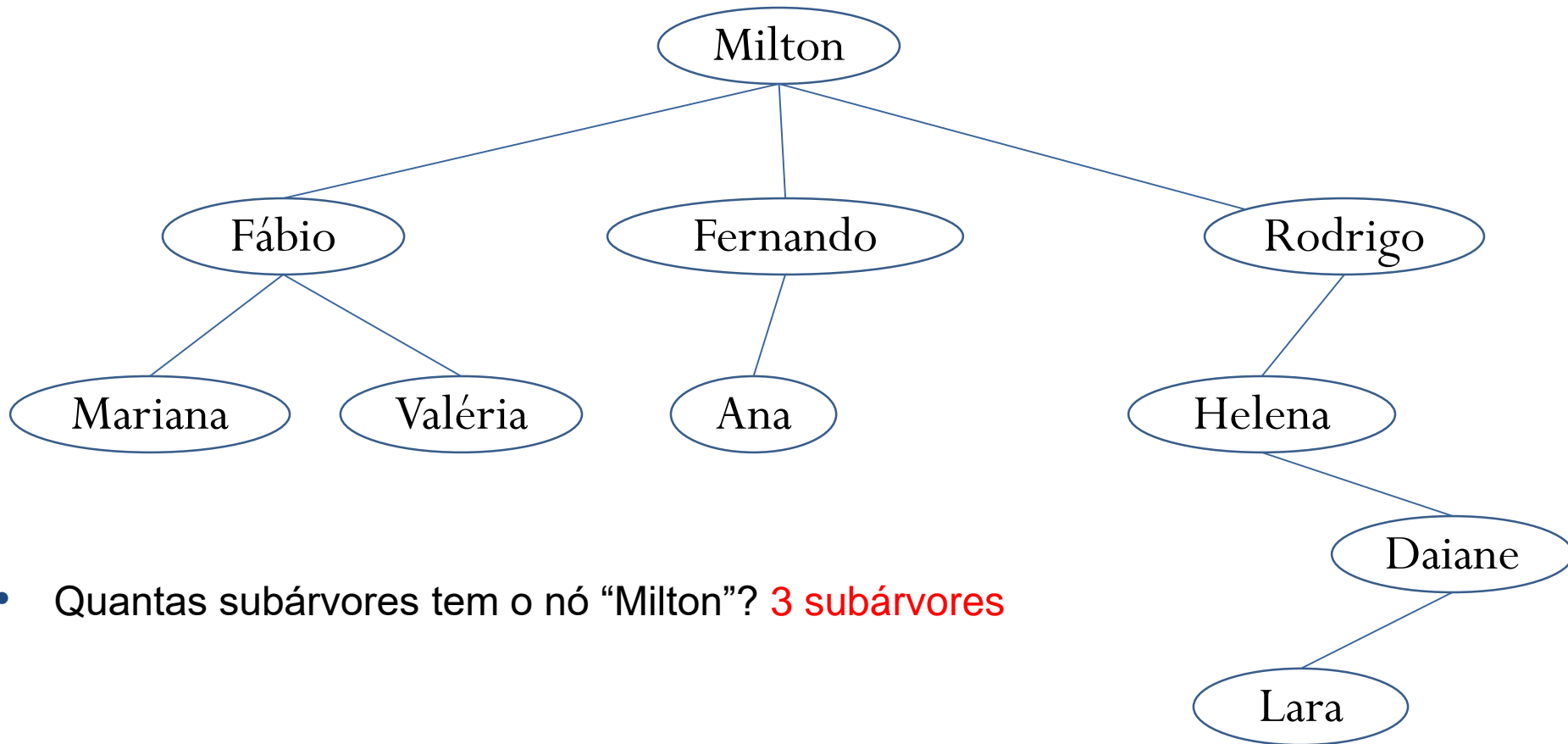


# Exemplo



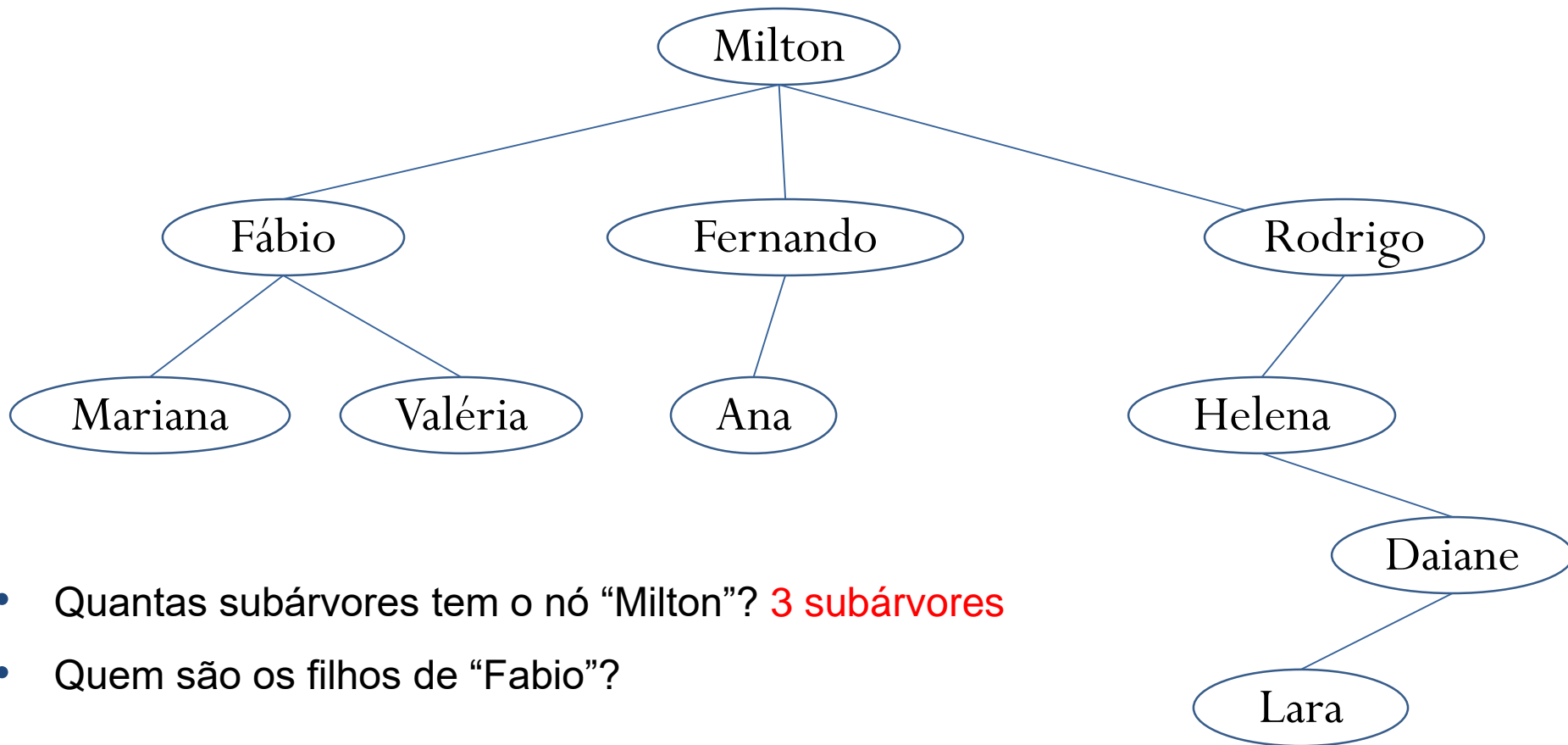
- Quantas subárvores tem o nó “Milton”?

# Exemplo



- Quantas subárvores tem o nó “Milton”? **3 subárvores**

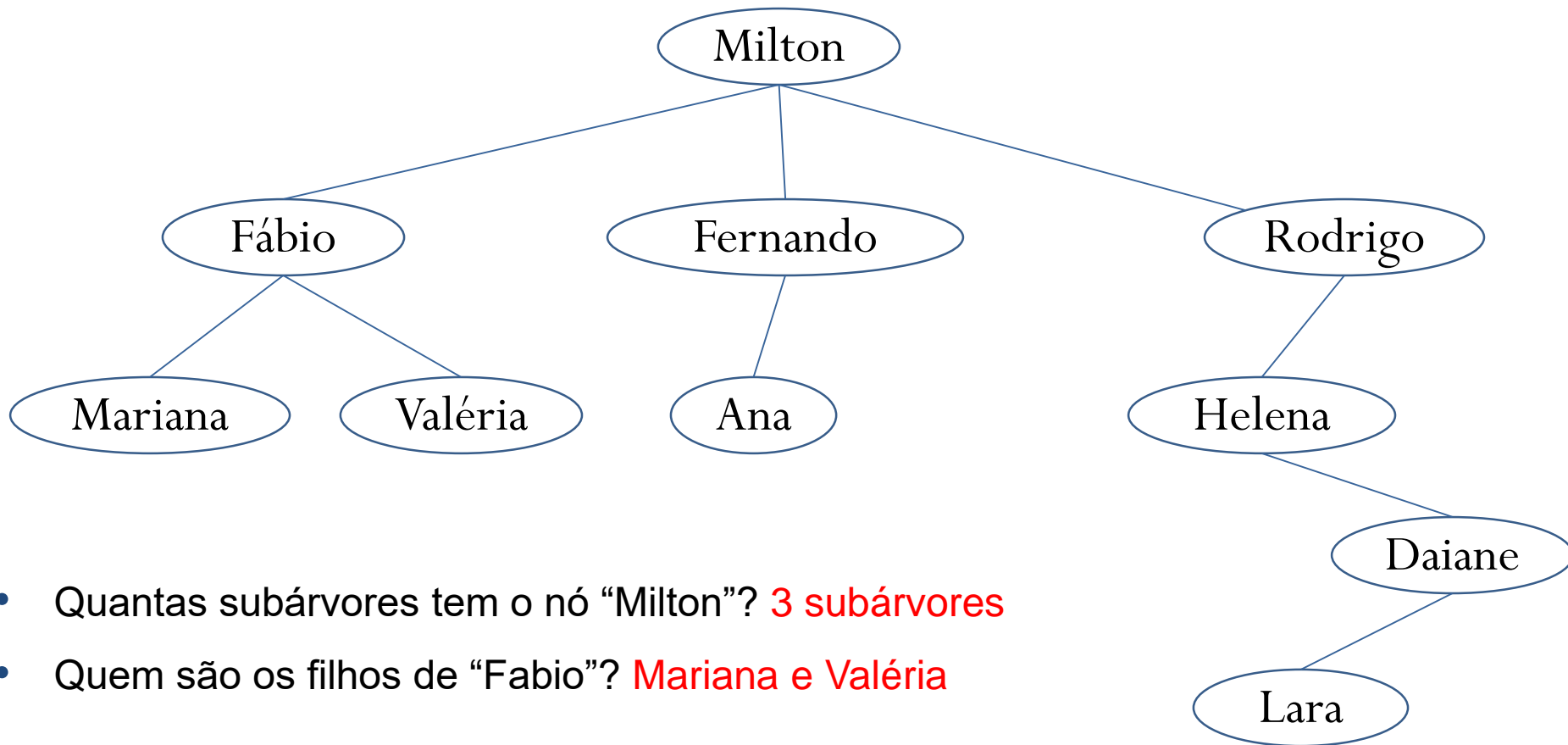
# Exemplo



- Quantas subárvores tem o nó “Milton”? **3 subárvores**
- Quem são os filhos de “Fábio”?



# Exemplo



- Quantas subárvores tem o nó “Milton”? **3 subárvores**
- Quem são os filhos de “Fábio”? **Mariana e Valéria**

# Exemplo



- Quantas subárvores tem o nó “Milton”? **3 subárvores**
- Quem são os filhos de “Fábio”? **Mariana e Valéria**
- Quem são os nós folhas?

# Exemplo



- Quantas subárvores tem o nó “Milton”? **3 subárvores**
- Quem são os filhos de “Fábio”? **Mariana e Valéria**
- Quem são os nós folhas? **Mariana, Valéria, Ana e Lara**

# Exemplo



- Quantas subárvores tem o nó “Milton”? **3 subárvores**
- Quem são os filhos de “Fábio”? **Mariana e Valéria**
- Quem são os nós folhas? **Mariana, Valéria, Ana e Lara**
- Quem são os descendentes de “Rodrigo”?

# Exemplo



- Quantas subárvores tem o nó “Milton”? **3 subárvores**
- Quem são os filhos de “Fábio”? **Mariana e Valéria**
- Quem são os nós folhas? **Mariana, Valéria, Ana e Lara**
- Quem são os descendentes de “Rodrigo”? **Helena, Daiane e Lara**

# Exemplo



- Quantas subárvores tem o nó “Milton”? **3 subárvores**
- Quem são os filhos de “Fábio”? **Mariana e Valéria**
- Quem são os nós folhas? **Mariana, Valéria, Ana e Lara**
- Quem são os descendentes de “Rodrigo”? **Helena, Daiane e Lara**
- Qual o grau dessa árvore?

# Exemplo



- Quantas subárvores tem o nó “Milton”? **3 subárvores**
- Quem são os filhos de “Fábio”? **Mariana e Valéria**
- Quem são os nós folhas? **Mariana, Valéria, Ana e Lara**
- Quem são os descendentes de “Rodrigo”? **Helena, Daiane e Lara**
- Qual o grau dessa árvore? **3**

# Exemplo



- Quantas subárvores tem o nó “Milton”? **3 subárvores**
- Quem são os filhos de “Fabio”? **Mariana e Valéria**
- Quem são os nós folhas? **Mariana, Valéria, Ana e Lara**
- Quem são os descendentes de “Rodrigo”? **Helena, Daiane e Lara**
- Qual o grau dessa árvore? **3**
- Qual a altura da árvore?



# Exemplo



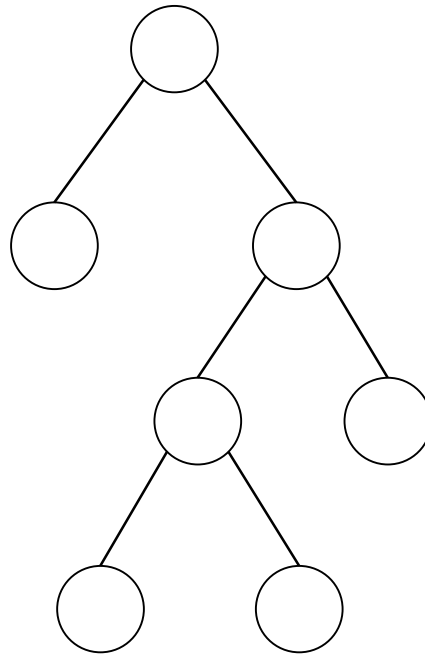
- Quantas subárvores tem o nó “Milton”? **3 subárvores**
- Quem são os filhos de “Fábio”? **Mariana e Valéria**
- Quem são os nós folhas? **Mariana, Valéria, Ana e Lara**
- Quem são os descendentes de “Rodrigo”? **Helena, Daiane e Lara**
- Qual o grau dessa árvore? **3**
- Qual a altura da árvore? **4**

# Árvores Binárias

- Uma Arvore Binária  $T$  é um conjunto finito de elementos, denominados nós ou vértices, tal que:
  - $T = \emptyset$ , a árvore binária é dita vazia; ou
  - $T = \{r\} \cup \{T_e\} \cup \{T_d\}$ .
- $T$  contém um nó especial  $r$ , chamado de raiz, e os demais nós podem ser subdivididos em dois subconjuntos distintos  $T_e$  e  $T_d$ , os quais também são árvores binárias
  - $T_e$ : subárvore à esquerda de  $T$
  - $T_d$ : subárvore à direita de  $T$

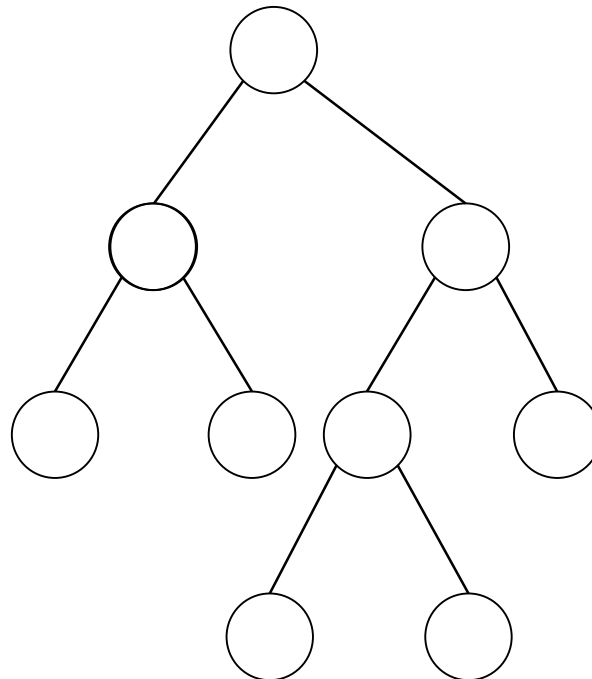
# Tipos de Árvores Binária

- Árvore estritamente binária
  - Todos os nós possuem 0 ou 2 filhos
  - Nós interiores sempre possuem 2 filhos



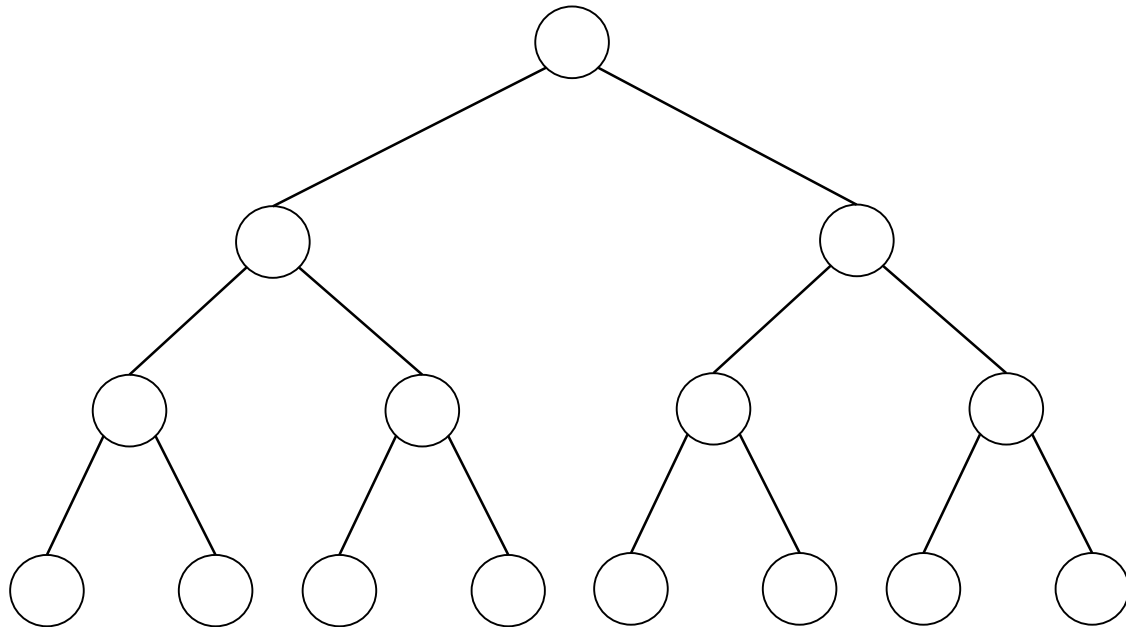
# Tipos de Árvores Binária

- Árvore Binária Completa
  - Se o nó  $v$  tem alguma subárvore vazia, então  $v$  está no último ou penúltimo nível da árvore



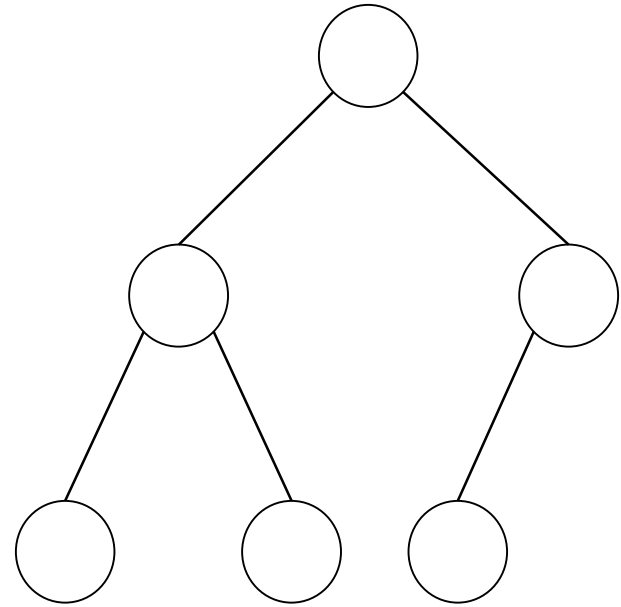
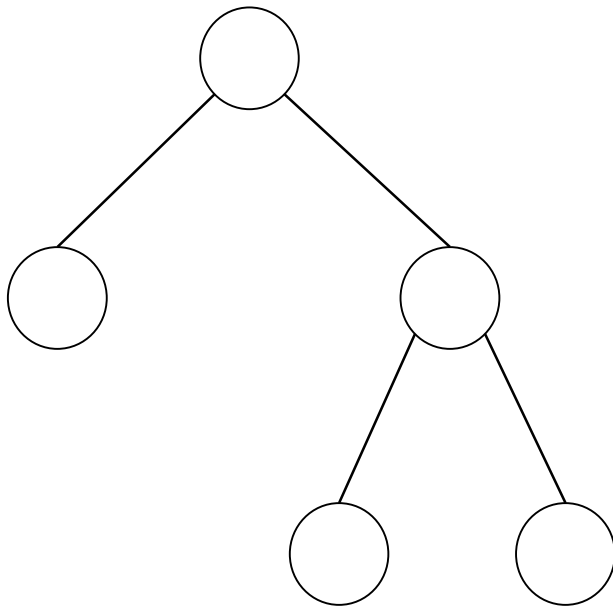
# Tipos de Árvores Binária

- Árvore Binária Cheia
  - Se o nó  $v$  tem alguma subárvore vazia, então  $v$  está no último nível da árvore
  - Obs.: uma árvore binária cheia é uma árvore binária completa e estritamente binária



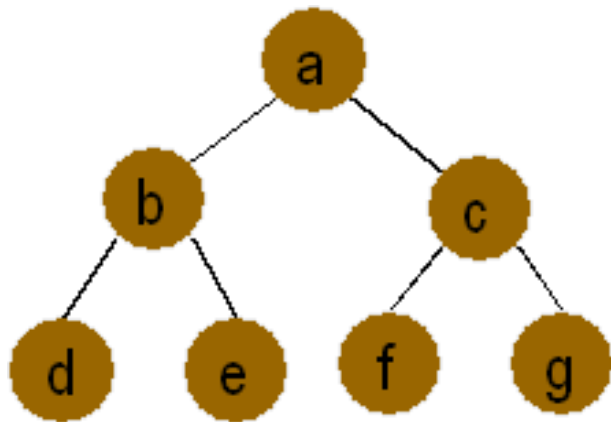
# Árvore Binária Balanceada

- Árvore Binária Balanceada
  - Para cada nó, as alturas de suas duas sub-árvores diferem de, no máximo, 1.



# Representação

- Representação estática



1	2	3	4	5	6	7	...
<b>a</b>	<b>b</b>	<b>c</b>	<b>d</b>	<b>e</b>	<b>f</b>	<b>g</b>	...

- Para um vetor indexado a partir da posição 1, se um nó está na posição  $i$ , seus filhos diretos estão nas posições
  - $2i$  : filho da esquerda
  - $2i+1$ : filho da direita

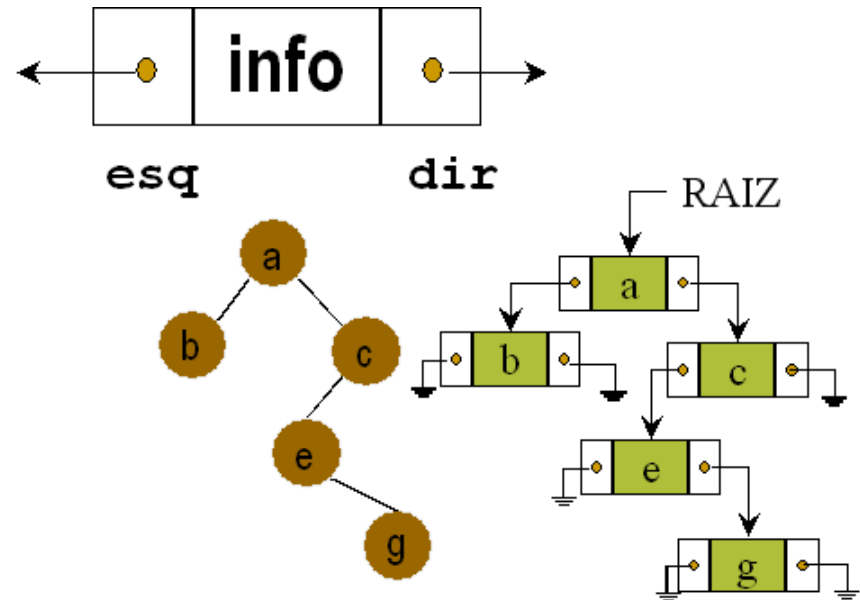
# Representação

- Representação dinâmica

```
struct ITEM{  
    int valor;  
};
```

```
struct NO{  
    ITEM item;  
    struct NO *fesq;  
    struct NO *fdir;  
};
```

```
struct ARVORE_BINARIA{  
    NO *raiz;  
};
```





# Estática x Dinâmica

	<b>Vantagens</b>	<b>Desvantagens</b>
<b>Estática</b>	Fácil implementação	Mau aproveitamento de memória
<b>Dinâmica</b>	Fácil manipulação	Gasto extra de memória para os ponteiros

# Operações básicas em Árvores Binárias

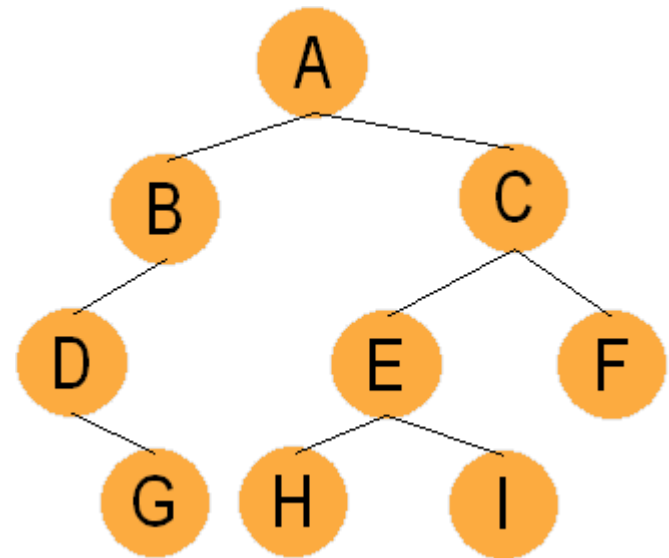
- Criar árvore
- Verificar se a árvore está vazia
- Imprimir elementos da árvore
- Determinar a altura da árvore
- Buscar um elemento
- Buscar pai de um elemento
- Inserir elemento à esquerda de outro elemento
- Inserir elemento à direita de outro elemento
- Finalizar árvore

# Percurso em Árvores Binárias

- Percorrer uma árvore binária “visitando” cada nó uma única vez
  - “Visitar” um nó pode ser:
    - Imprimir seu valor armazenado
    - Alterar o valor
    - etc
- Não existe um único percurso para árvores (binárias ou não): diferentes percursos podem ser realizados, dependendo da aplicação.
- 3 percursos básicos: pré-ordem, em-ordem e pós-ordem

# Percurso em Árvores Binárias

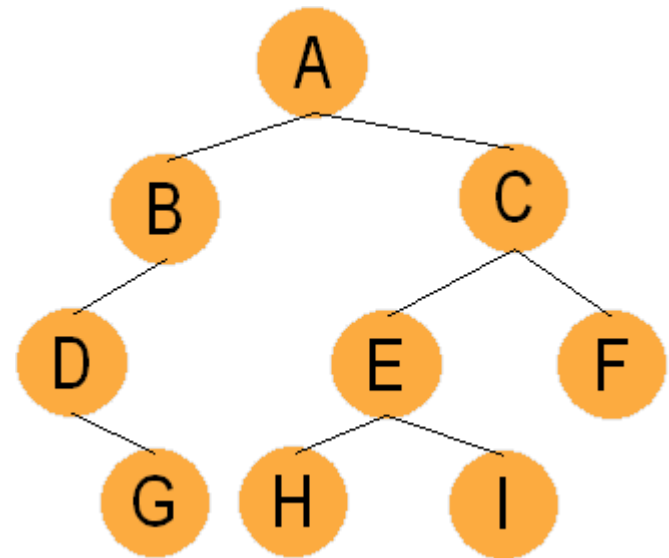
- Pré-ordem
  - Visita o nó raiz
  - Percorre a subárvore esquerda
  - Percorre a subárvore direita



Resultado: ABDGCEHIF

# Percurso em Árvores Binárias

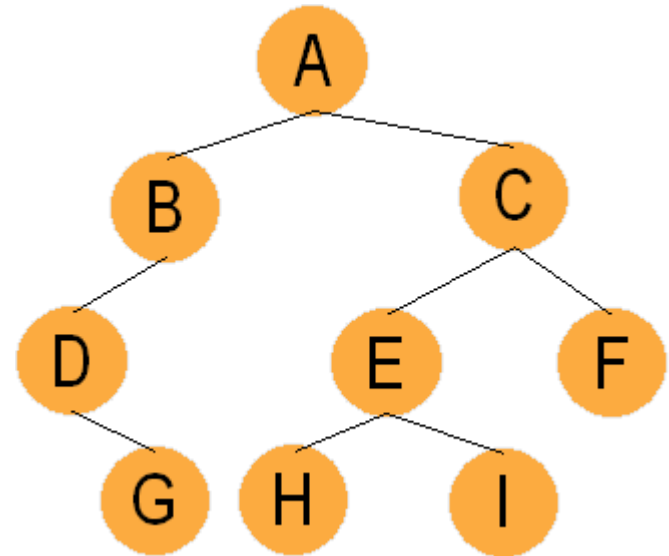
- Em-ordem
  - Percorre a subárvore esquerda
  - Visita o nó raiz
  - Percorre a subárvore direita



Resultado: DGBAHEICF

# Percurso em Árvores Binárias

- Pós-ordem
  - Percorre a subárvore esquerda
  - Percorre a subárvore direita
  - Visita o nó raiz



Resultado: GDBHIEFCA