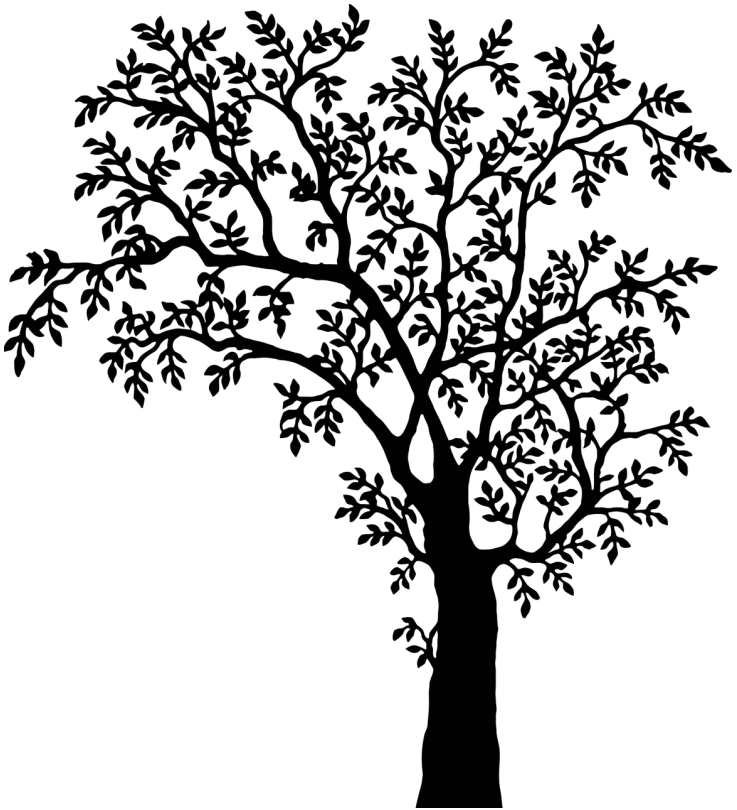
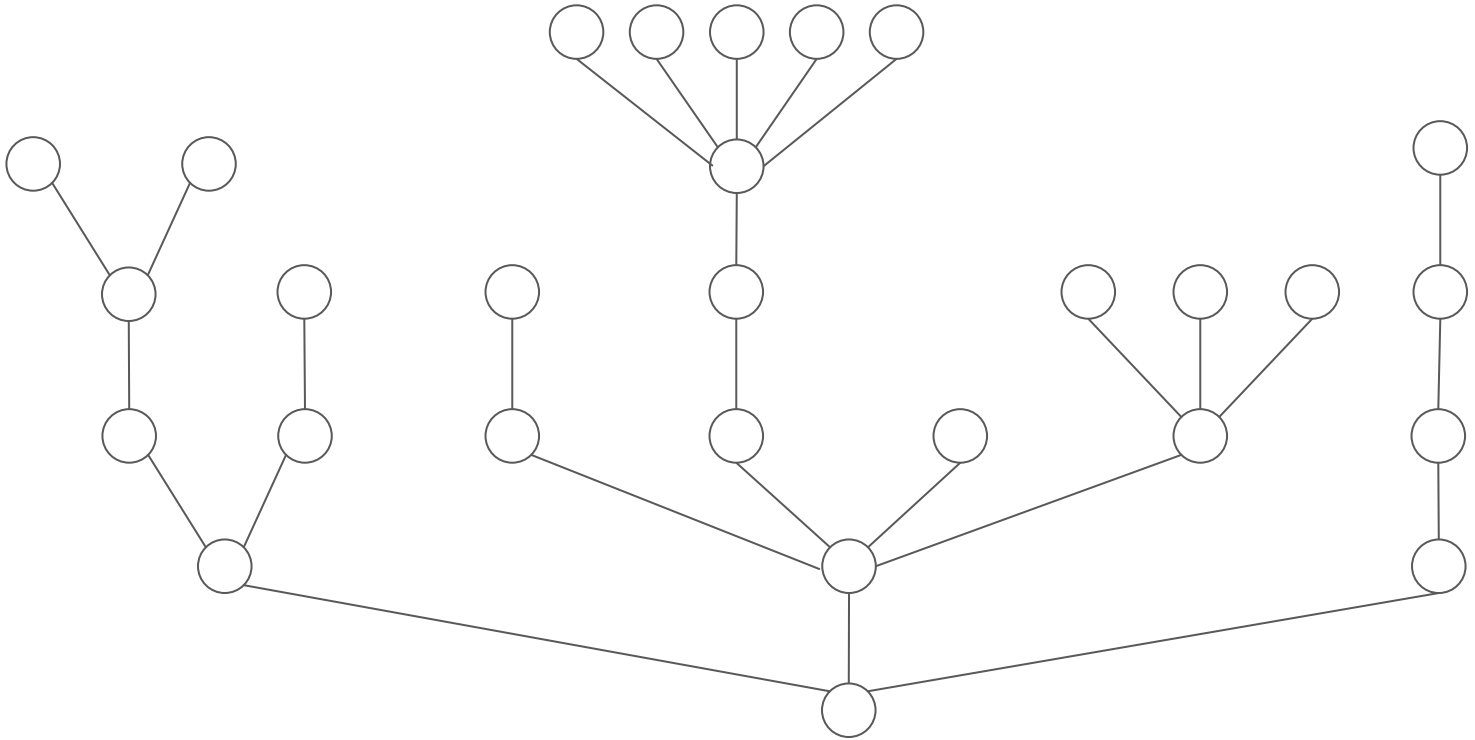
Árvores

Árvore

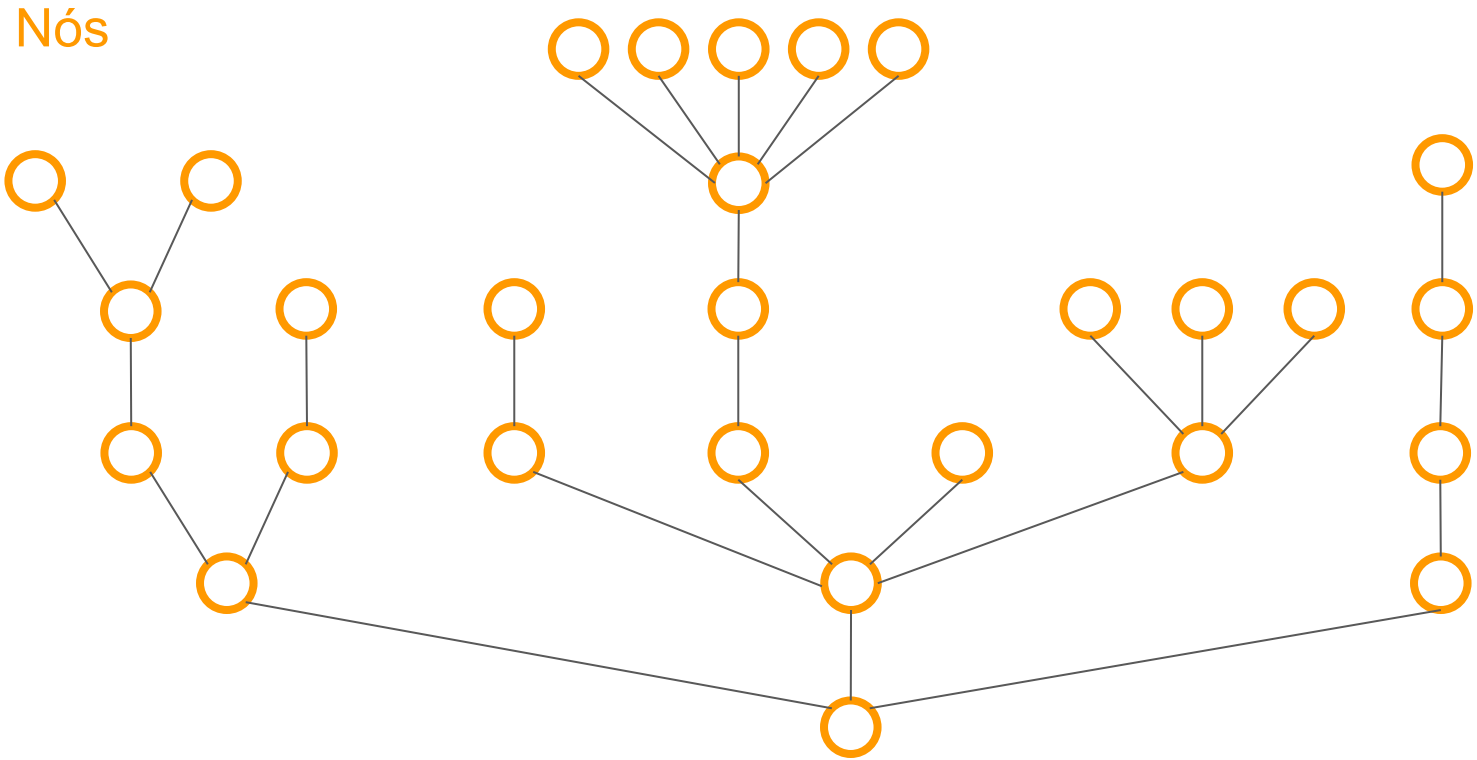
Abstraindo a estrutura de uma árvore



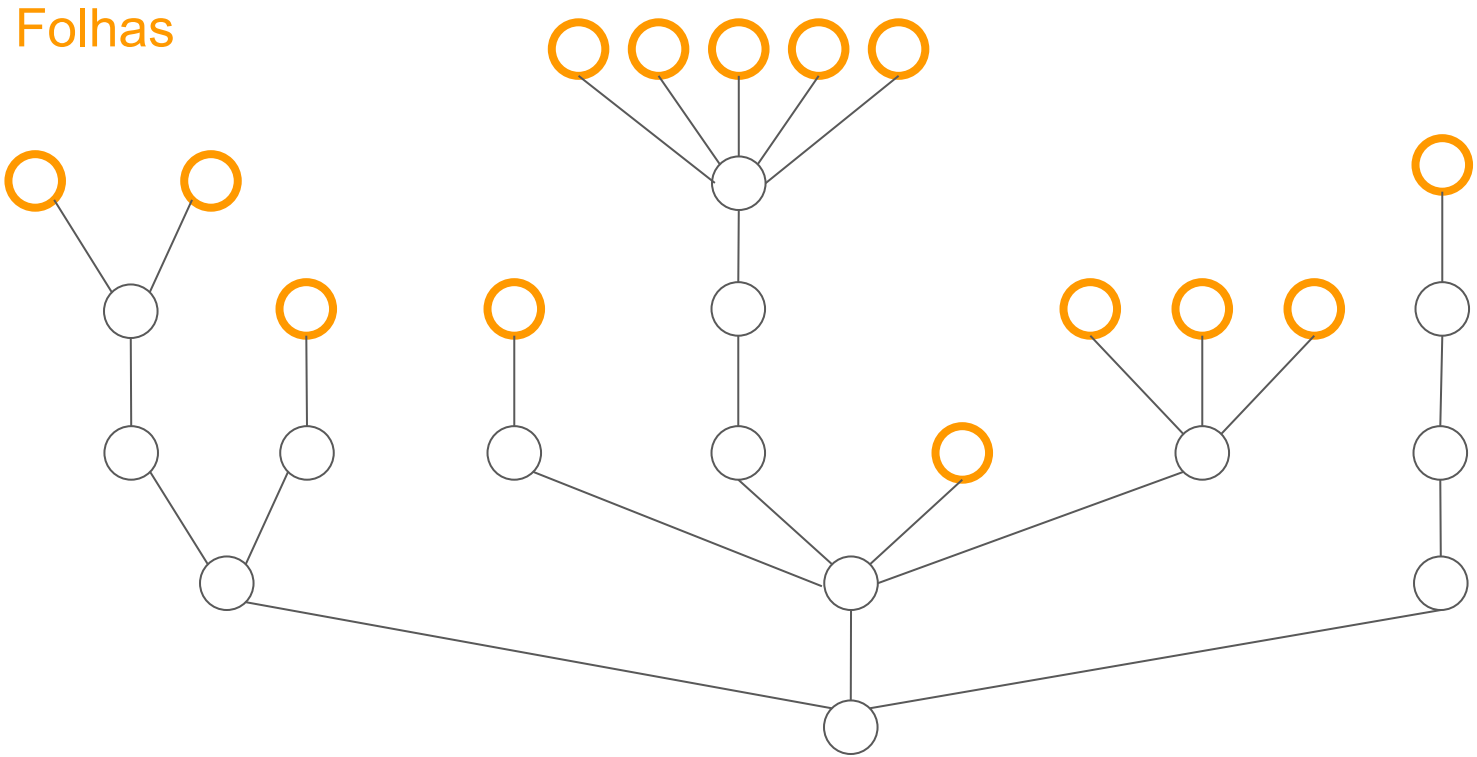
Abstraindo a estrutura de uma árvore



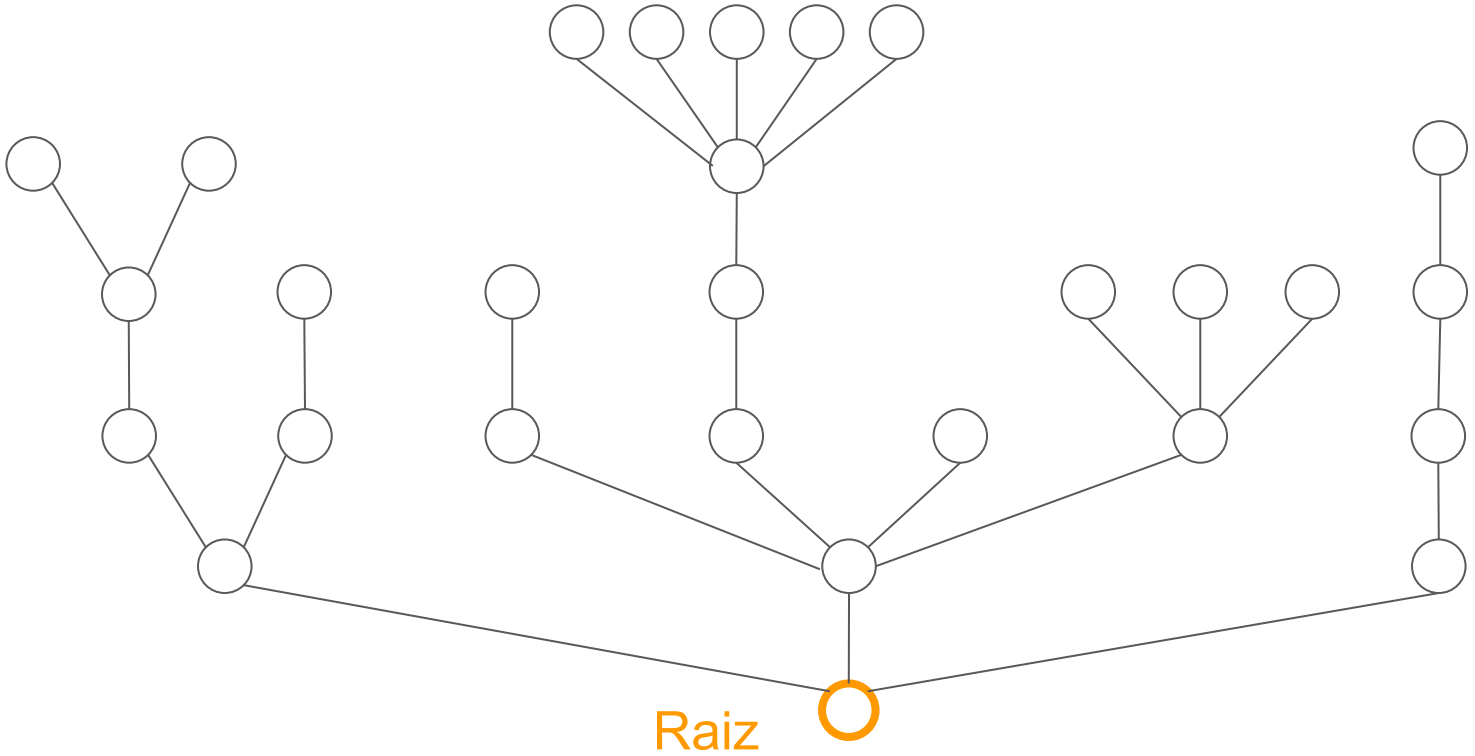
Abstraindo a estrutura de uma árvore



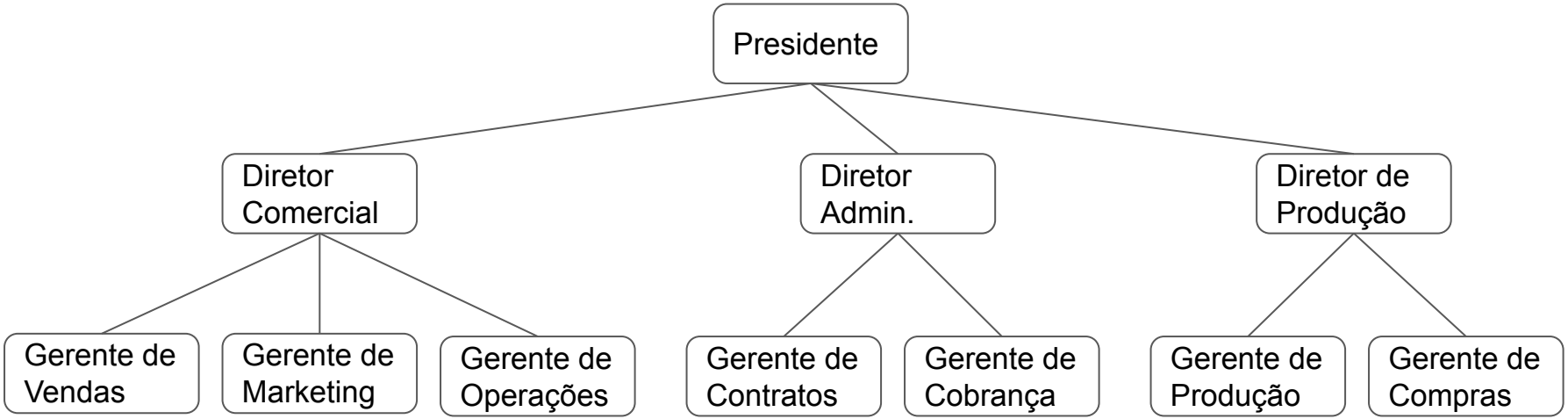
Abstraindo a estrutura de uma árvore



Abstraindo a estrutura de uma árvore

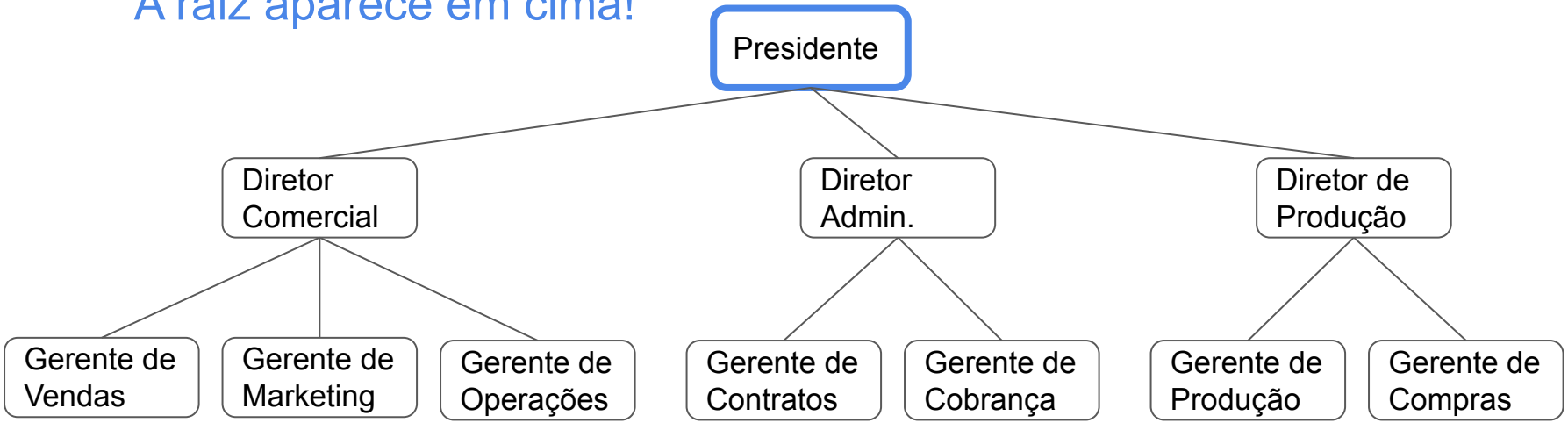


Exemplo de uso: organograma de uma empresa

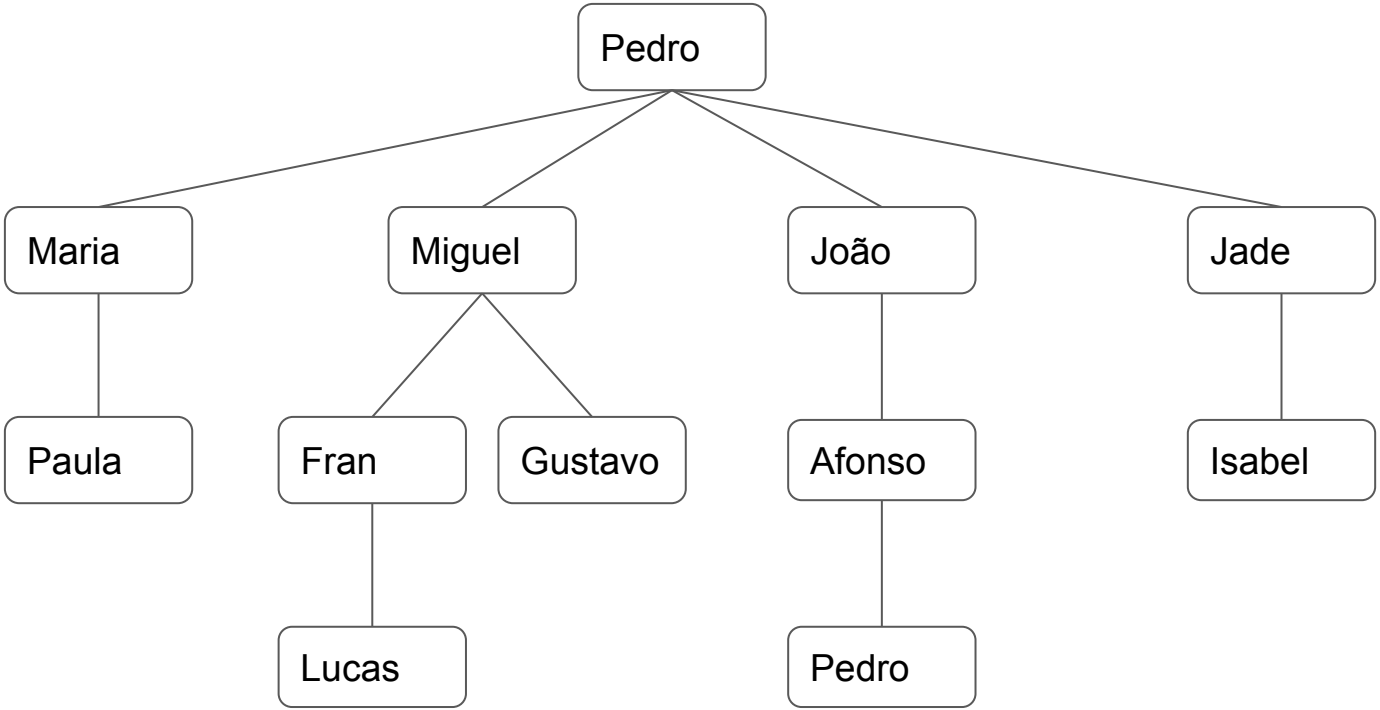


Exemplo de uso: organograma de uma empresa

A raiz aparece em cima!



Exemplo de uso: descendência de uma pessoa



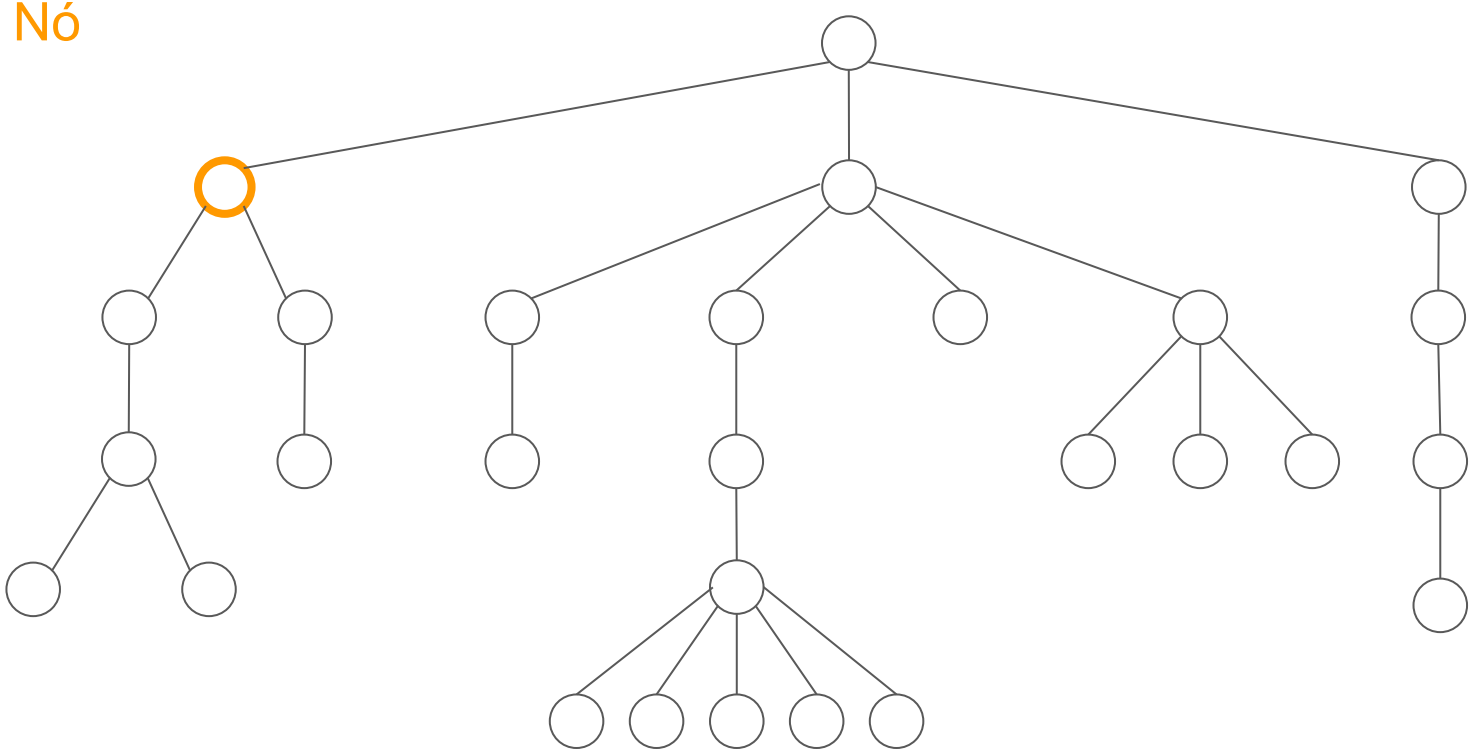
Árvores - uso na Computação

- Árvores são amplamente utilizadas na Computação
  - como uma ferramenta para descrever propriedades de algoritmos e
  - como uma estrutura de dados de fato

Árvore - definição

- Um **nó** (ou vértice) contém uma informação útil

Árvore - definição

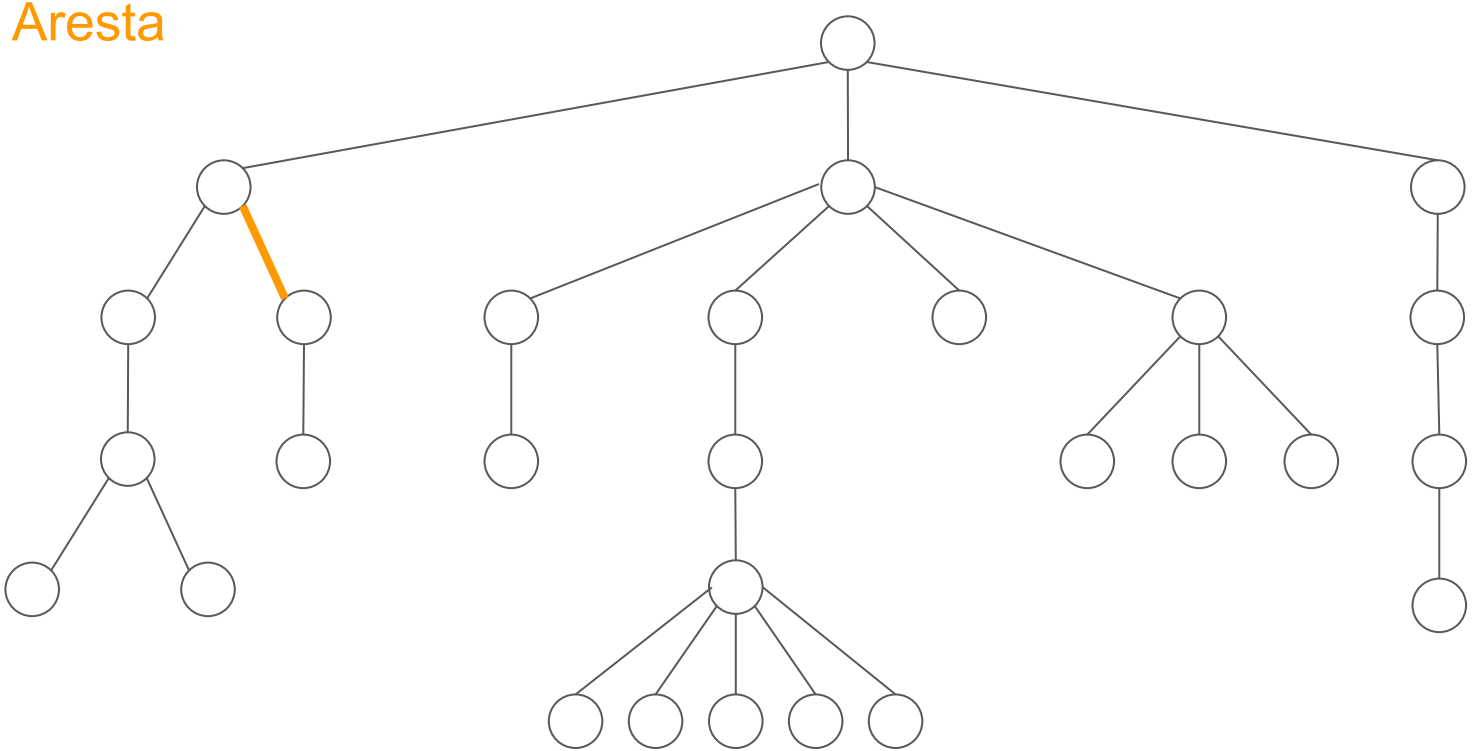


Árvore - definição

- Um **nó** (ou vértice) contém uma informação útil
- Uma **aresta** é uma conexão entre dois nós

Árvore - definição

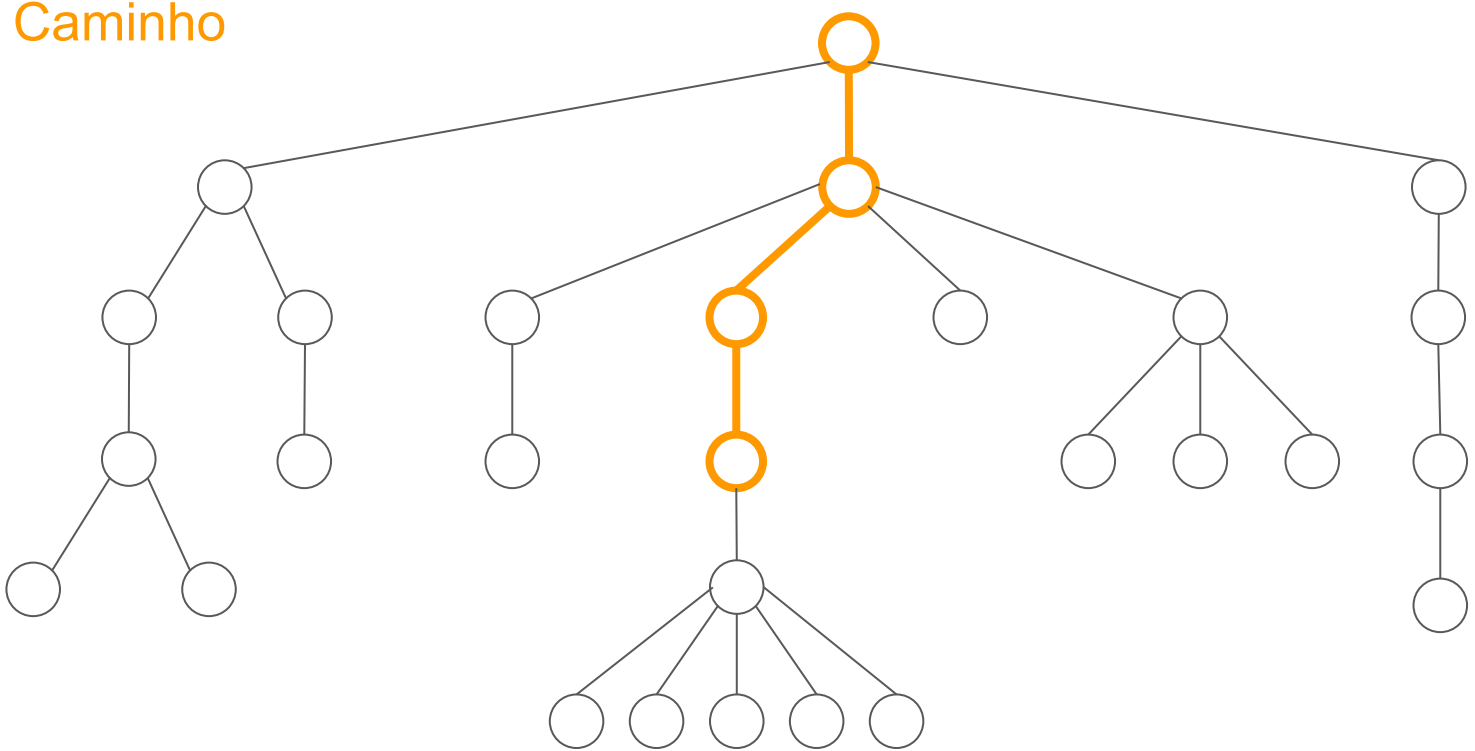
Aresta



Árvore - definição

- Um **nó** (ou vértice) contém uma informação útil
- Uma **aresta** é uma conexão entre dois nós
- Um **caminho** é uma sequência de nós $< v_0, v_1, v_2, \ldots, v_k >$ tal que
  - todos os nós da sequência são distintos e
  - existe uma aresta conectando v_{i-1} e v_i para i = 1, 2, …, k

Árvore - definição

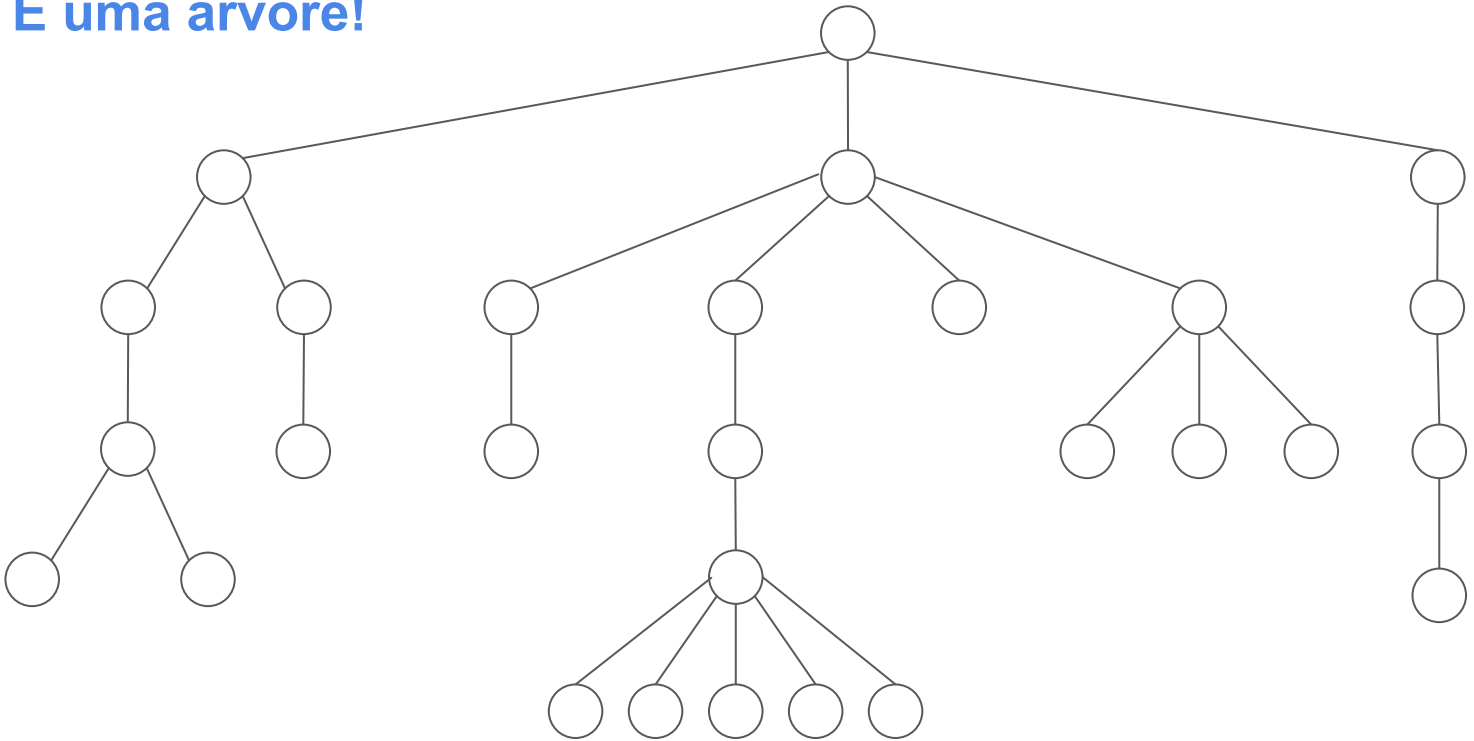


Árvore - definição

- Um **nó** (ou vértice) contém uma informação útil
- Uma **aresta** é uma conexão entre dois nós
- Um **caminho** é uma sequência de nós $< v_0, v_1, v_2, \ldots, v_k >$ tal que
  - todos os nós da sequência são distintos
  - existe uma aresta conectando v_{i-1} e v_i para $i = 1, 2, \ldots, k$
- Uma **árvore** é formada por nós e arestas tais que
  - existe **exatamente um caminho** conectando cada par de vértices

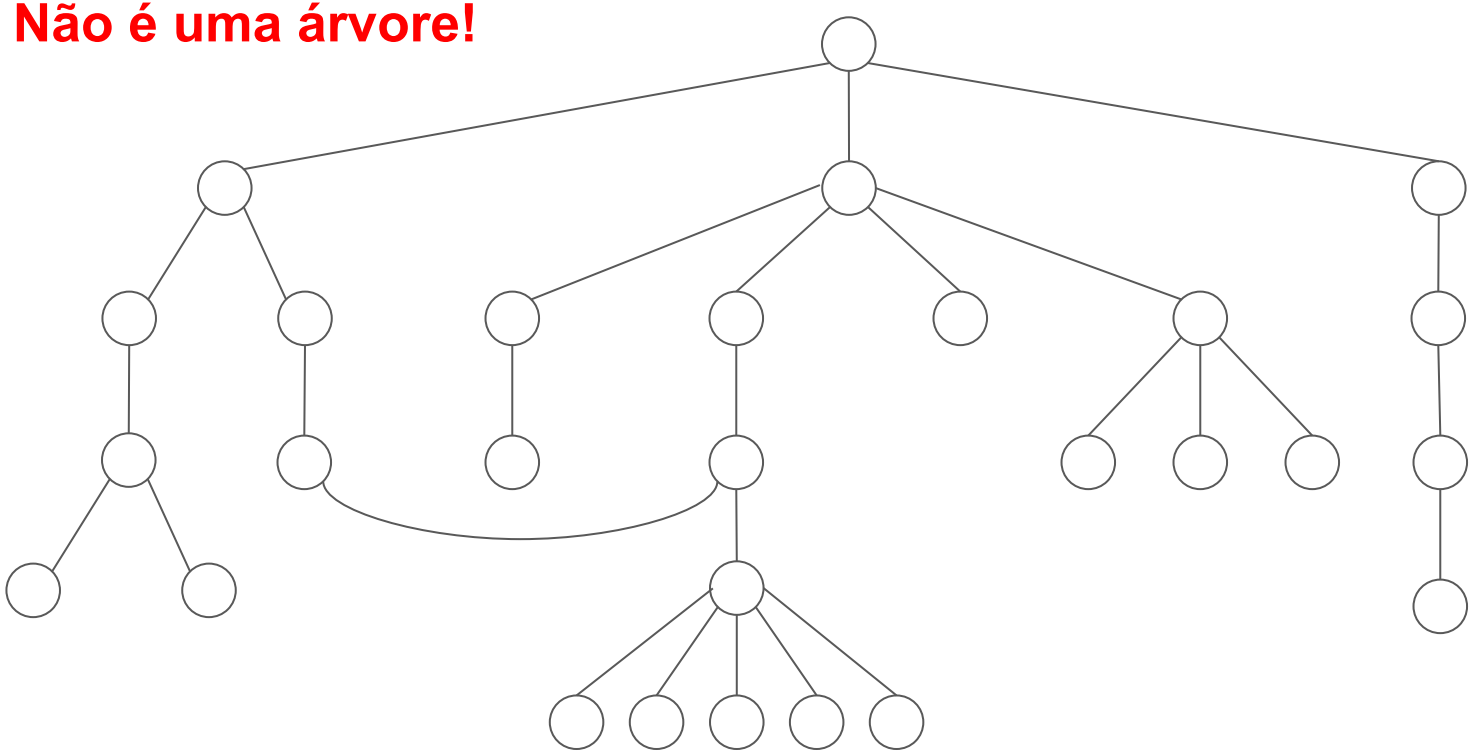
Árvore - definição

É uma árvore!



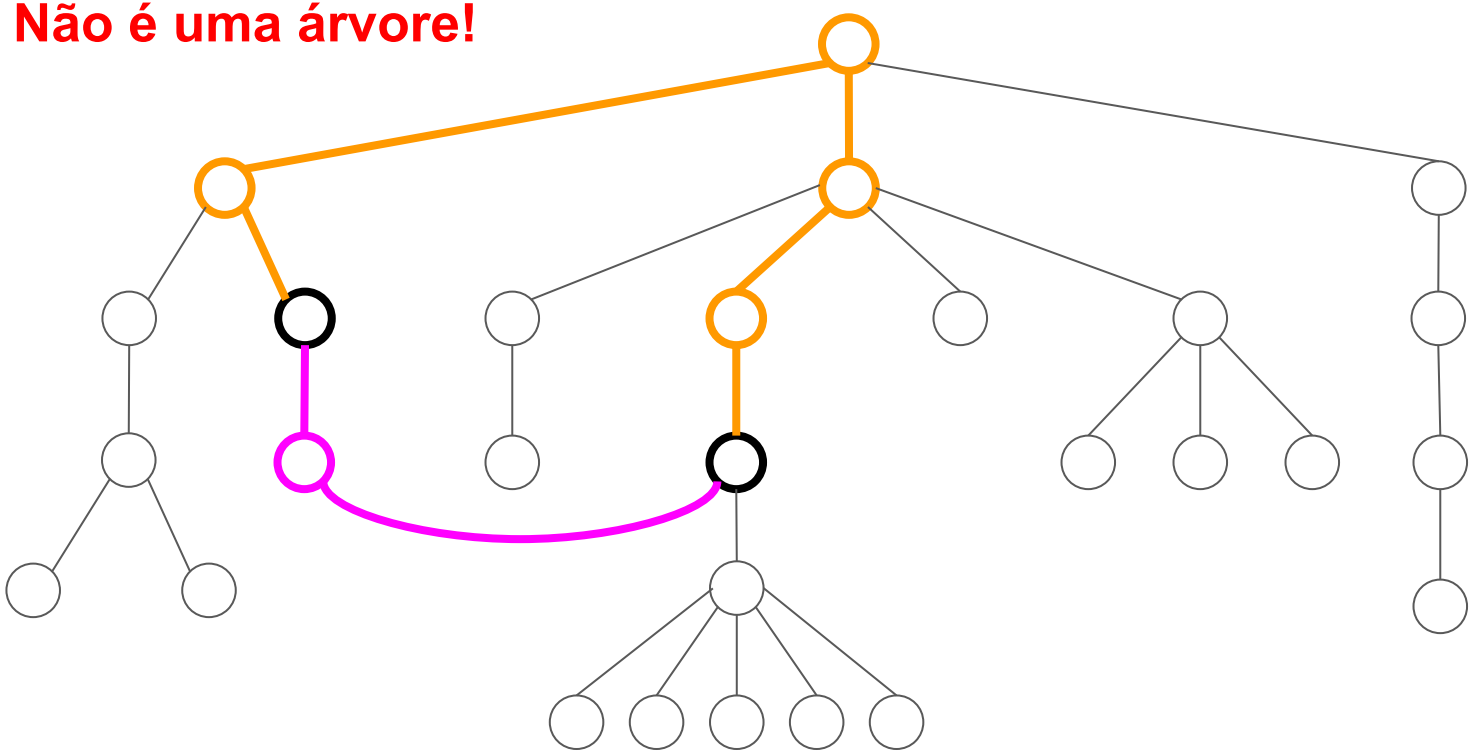
Árvore - definição

Não é uma árvore!



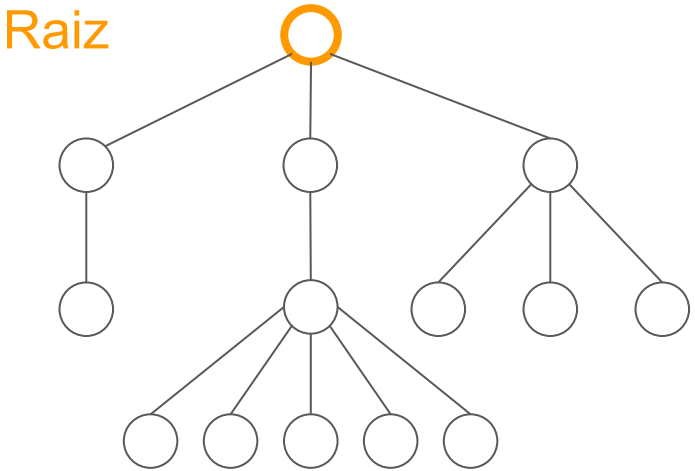
Árvore - definição

Não é uma árvore!

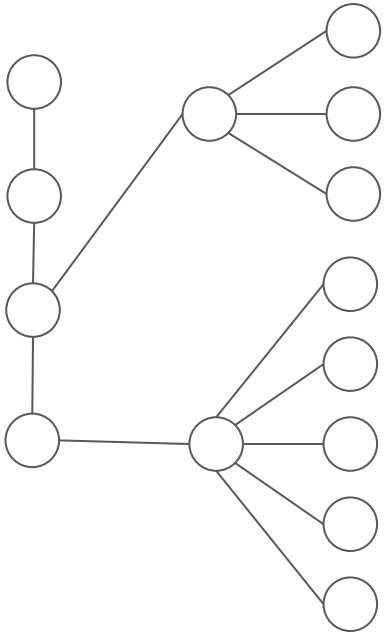


Árvore enraizada - terminologia

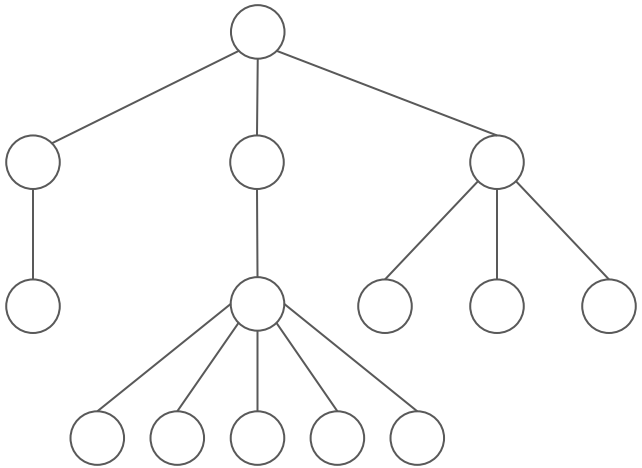
Árvore enraizada



Árvore não enraizada

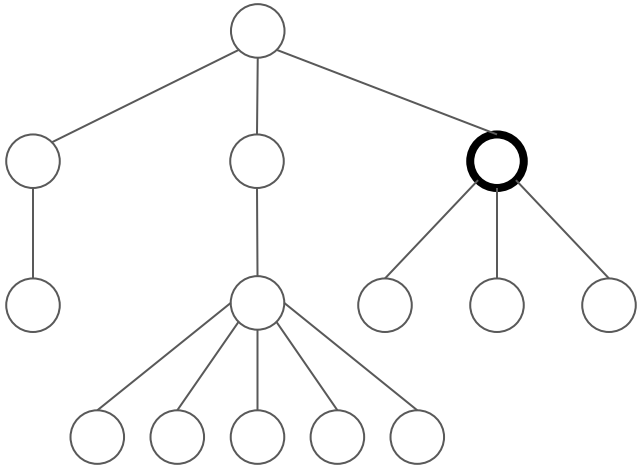


Árvore enraizada - terminologia



Árvore enraizada - terminologia

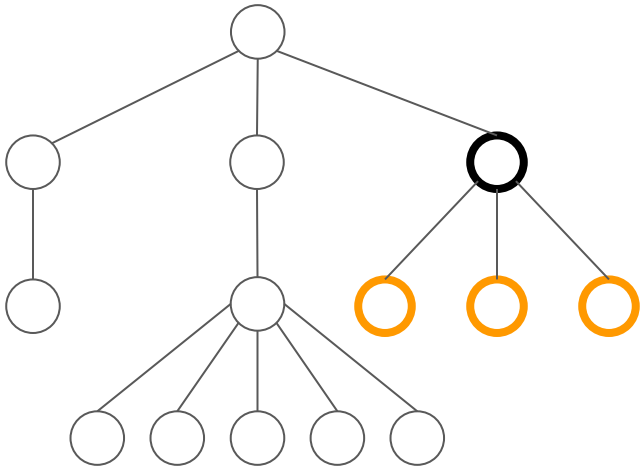
Nó



Árvore enraizada - terminologia

Nó

Filhos

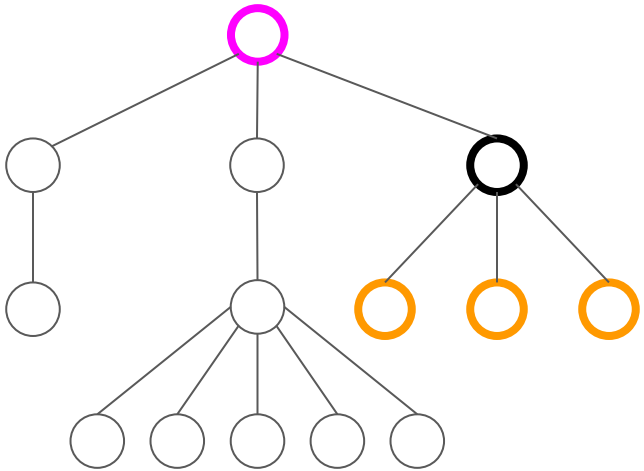


Árvore enraizada - terminologia

Pai

Nó

Filhos

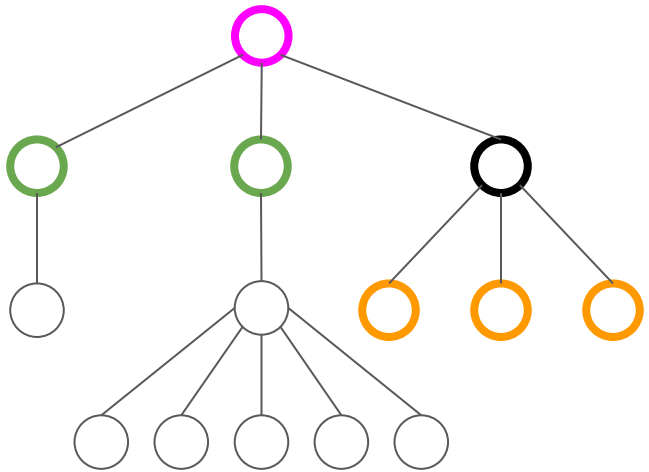


Árvore enraizada - terminologia

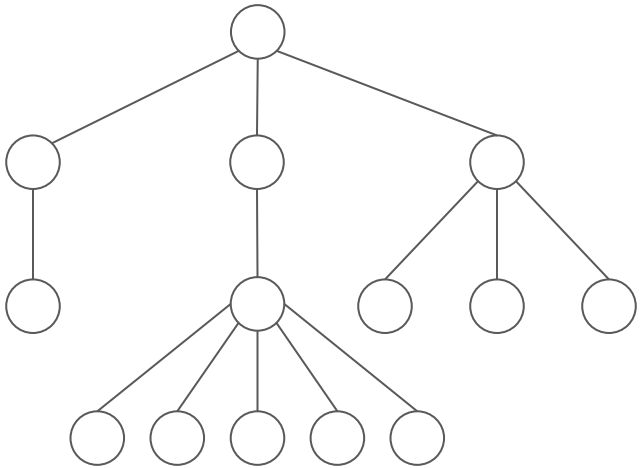
Pai

Nó Irmãos

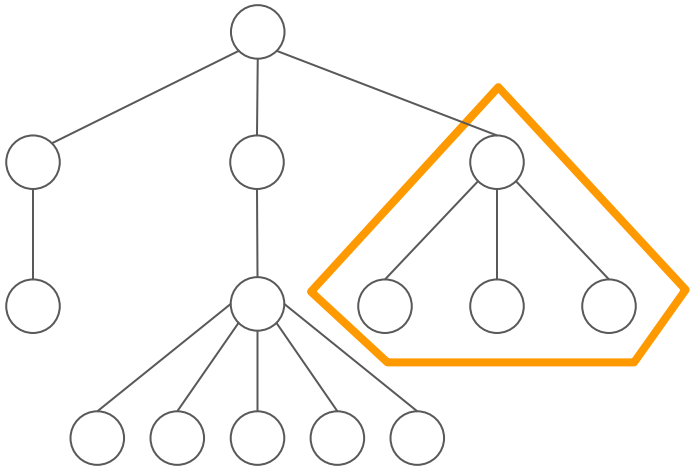
Filhos



Árvore enraizada - terminologia

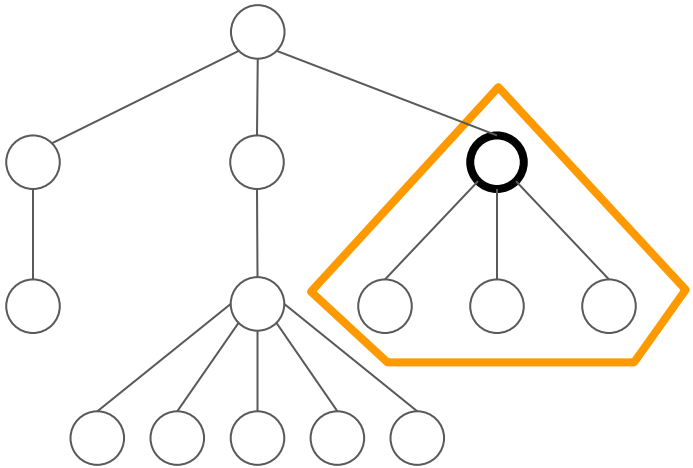


Árvore enraizada - terminologia



Subárvore

Árvore enraizada - terminologia



Raiz da subárvore

Subárvore

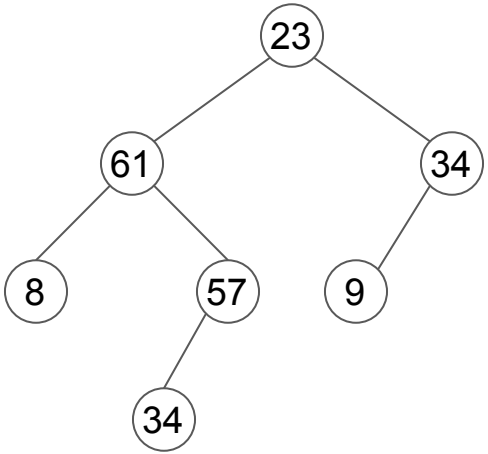
Árvore binária

- (Definição recursiva.) Uma **árvore binária**
  - ou não contém nenhum nó
  - ou é composta de um nó conectado:
    - a uma árvore binária chamada de **subárvore esquerda** do nó e
    - a uma árvore binária chamada de **subárvore direita** do nó

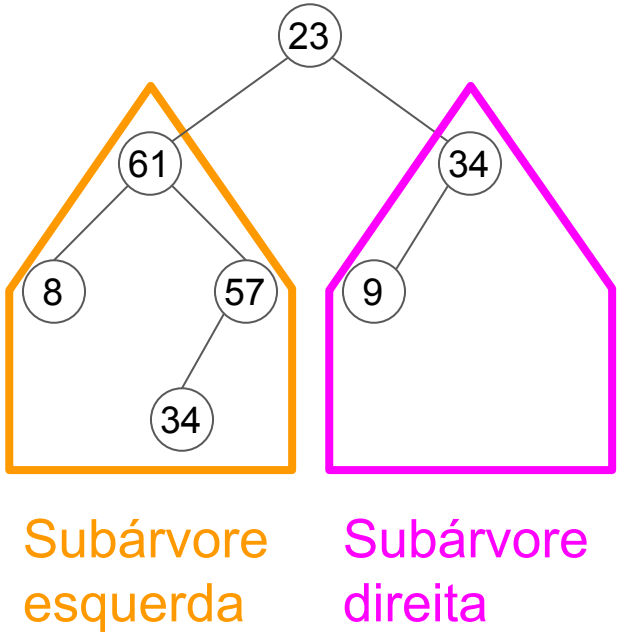
Árvore binária

- (Definição recursiv
  - ou não contém
  - ou é composta
    - a uma árv
    - a uma árv

Árvore binária

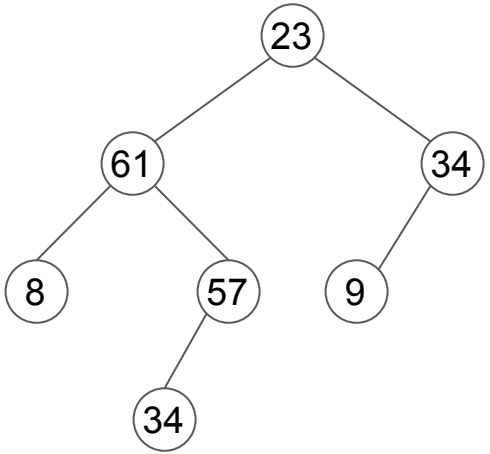


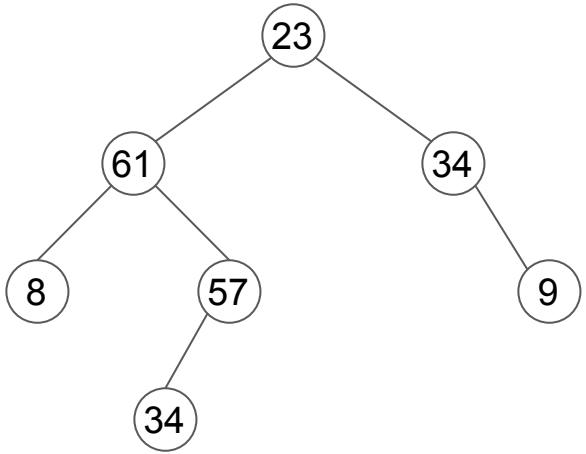
Árvore binária



Árvore binária

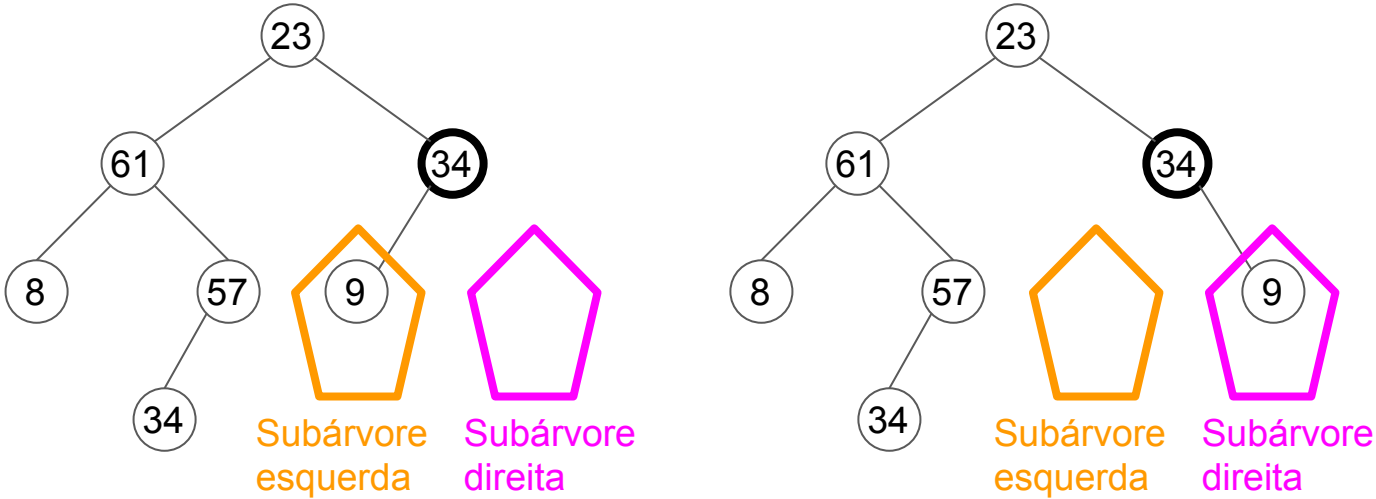
Estas árvores binárias são iguais?





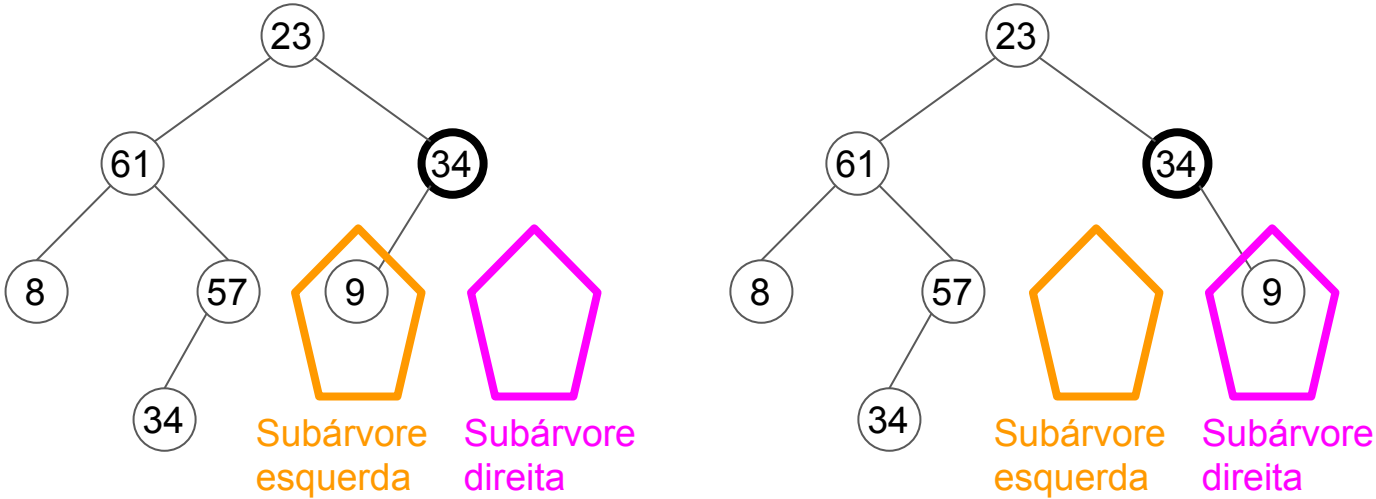
Árvore binária

Estas árvores binárias são iguais?



Árvore binária

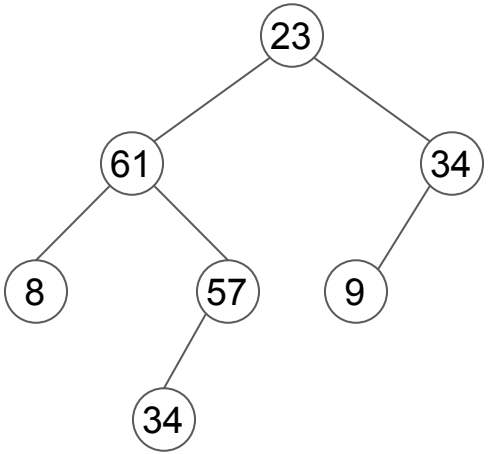
Estas árvores binárias são iguais? Não!



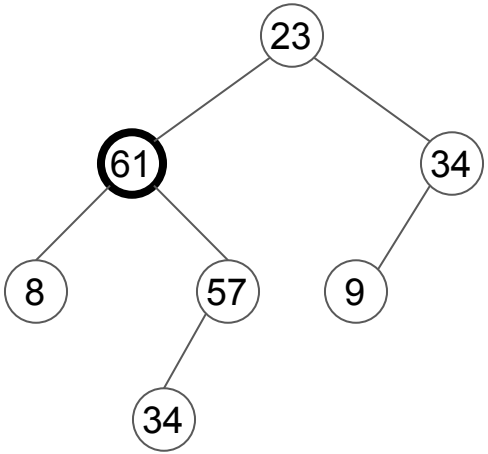
Árvore binária

- (Definição recursiva.) Uma **árvore binária**
  - ou não contém nenhum nó
  - ou é composta de um nó conectado:
    - a uma árvore binária chamada de **subárvore esquerda** do nó e
    - a uma árvore binária chamada de **subárvore direita** do nó
- Em uma árvore binária, um filho de um nó é chamado de
  - **filho esquerdo** se ele é a raiz da subárvore esquerda do nó e
  - **filho direito** se ele é a raiz da subárvore direita do nó

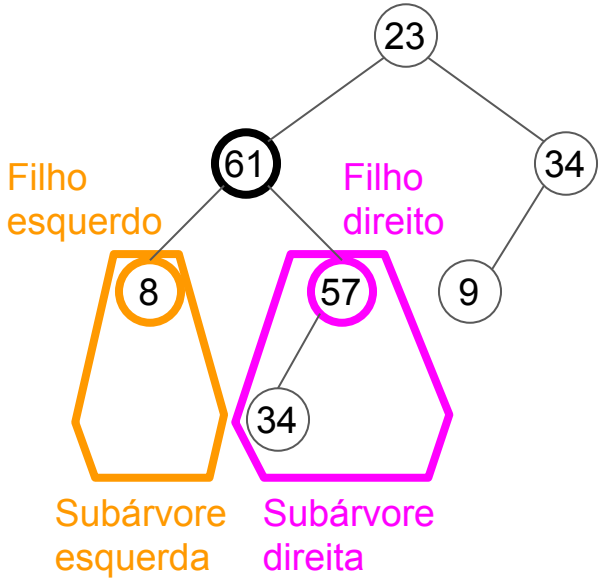
Árvore binária



Árvore binária



Árvore binária



Implementação de árvores binárias

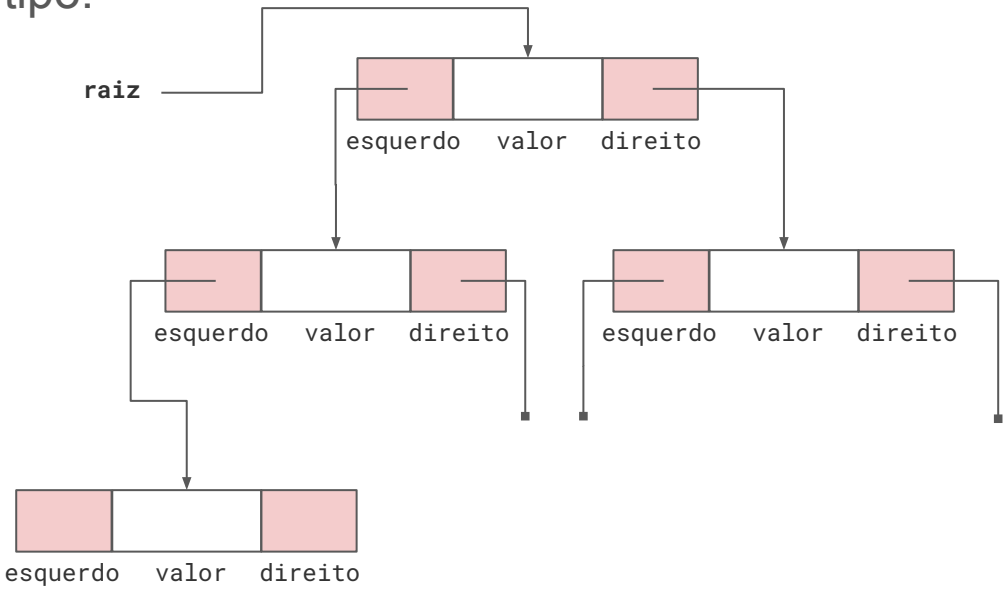
- Uma árvore binária é comumente implementada através de uma estrutura encadeada
- Esta estrutura é composta de nodos, cada um representando um nó da árvore
- Cada nodo é composto
  - pela informação do nó,
  - por um ponteiro para o seu filho esquerdo e
  - por um ponteiro para o seu filho direito

Implementação de árvores binárias

- Podemos declarar o seguinte tipo:

```c
typedef struct no {
    int valor;
    struct no *esquerdo;
    struct no *direito;
} No;
```

```c
No *raiz;
```



Exercícios

1. Desenhe uma árvore binária cuja estrutura seja baseada na tabela abaixo. Considere que a raiz da árvore é o nó de índice 4.

Nós			
Índice	**Dado**	**Filho esquerdo (índice)**	**Filho direito (índice)**
0	12	5	1
1	4	7	-
2	10	3	6
3	2	-	-
4	18	0	2
5	7	-	-
6	21	-	-
7	5	-	-

Exercícios

2. Escreva uma função recursiva
   `No *constroiArvBin(int indRaiz, int dado[], int filhoEsq[], int filhoDir[])`
   que recebe por parâmetro as informações descritas no exercício anterior e constrói uma árvore binária utilizando o tipo `No` definido nesta apresentação. A função deve retornar um ponteiro para a raiz da árvore binária.

https://codeshare.io/yoRY30

Referências

- Esta apresentação é baseada nas seguintes referências:
  - Capítulo 5 do livro
    Sedgewick, R., Algorithms in C, Addison-Wesley, 1998.
  - Capítulo 10 e Apêndice B.5 do livro
    Cormen, T., Leiserson, C., Rivest, R., Stein, C., Introduction To Algorithms, MIT Press, 2001.