Aula 2.1: Árvores - Introdução

DCC405-Estrutura de Dados II

Prof. Me. Acauan C. Ribeiro

Roteiro

- Definir árvores como estruturas de dados
- Definir os termos associados às árvores
- Discutir algoritmos de percurso (travessia) de árvores
- Discutir uma implementação de árvore binária
- Examinar um exemplo de árvore binária

Árvores

Introdução

- Semenstre passado vimos o conceito de **Lista Ligada (Linked List)**

Lista Liagada

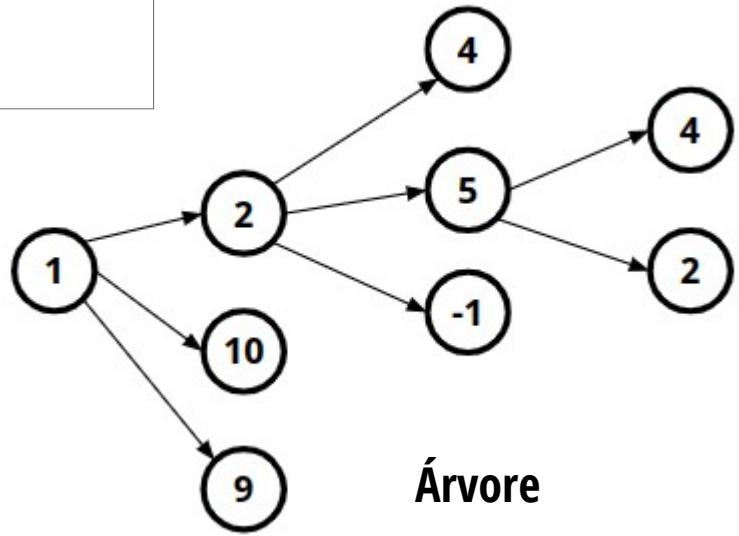
Introdução

- Semenstre passado vimos o conceito de **Lista Ligada (Linked List)**

Lista Liagada

```
struct Node {
      int data;
      Node* next;
}
```

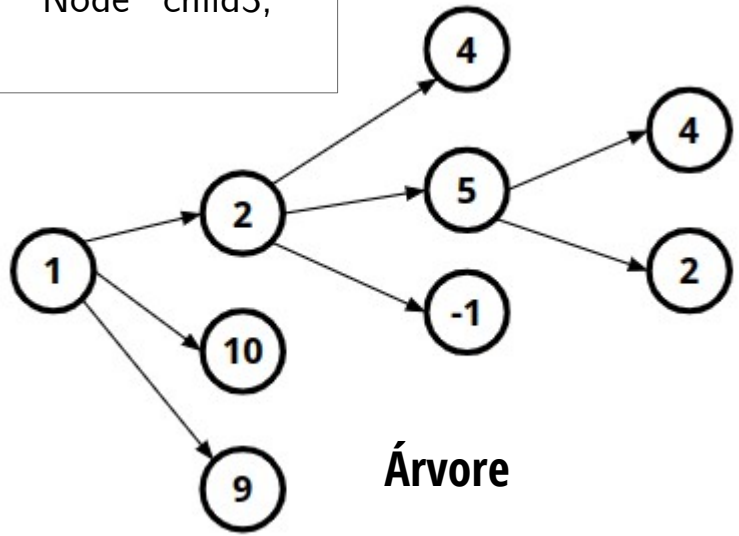
Introdução



Árvore

Introdução

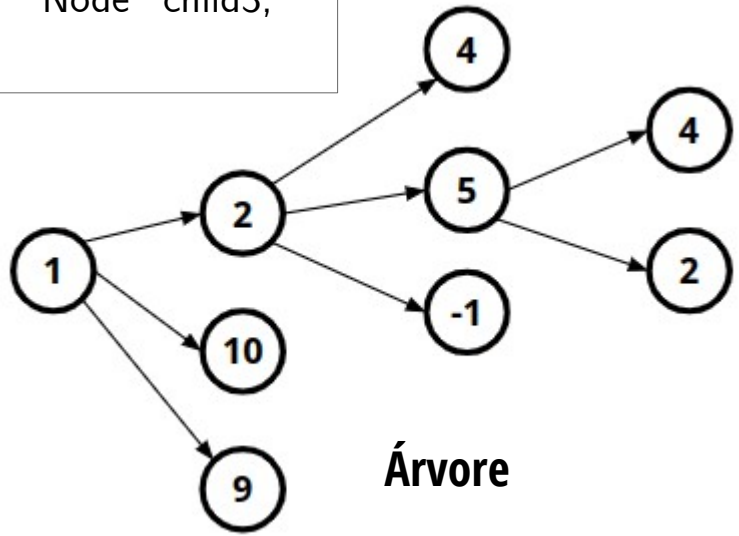
```
struct Node {
        int data;
        Node* child1;
        Node* child2;
        Node* child3;
}
```



Árvore

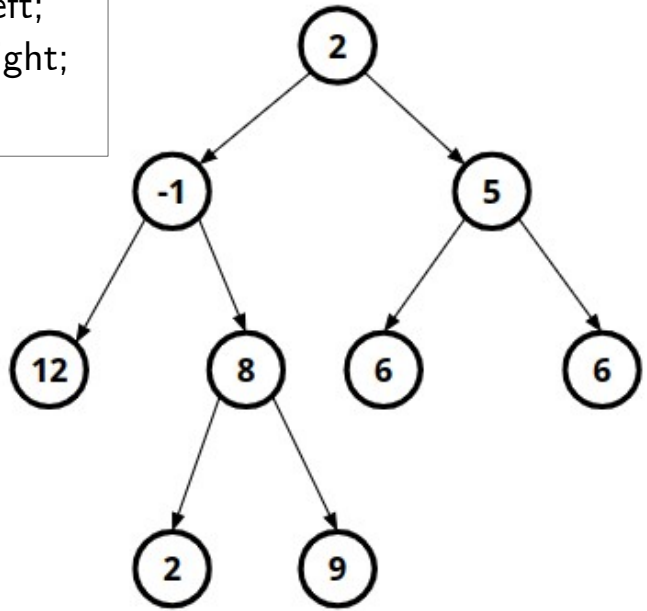
Introdução

```
struct Node {
        int data;
        Node* child1;
        Node* child2;
        Node* child3;
}
```



Árvore

```
struct Node {
        int data;
        Node* left;
        Node* right;
}
```



Árvore Binária

Game: É arvore ou não é?

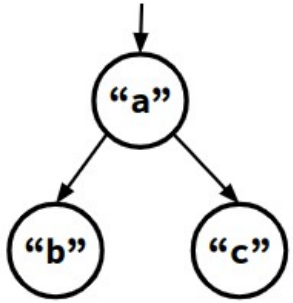
1 ?

2 ?

3 ?

4 ?

Game: É arvore ou não é?



?

2

?

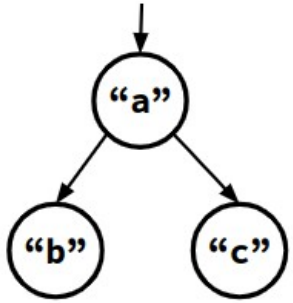
3

?

4

?

Game: É arvore ou não é?



True ✔

2

?

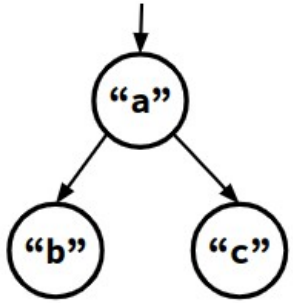
3

?

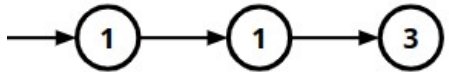
4

?

Game: É arvore ou não é?



True ✔



?

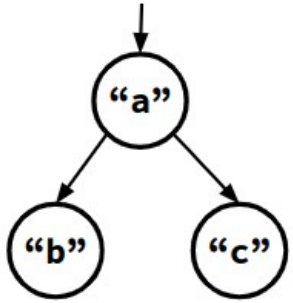
3

?

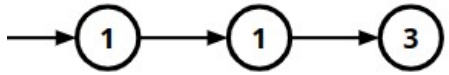
4

?

Game: É arvore ou não é?



True ✔



True ✔

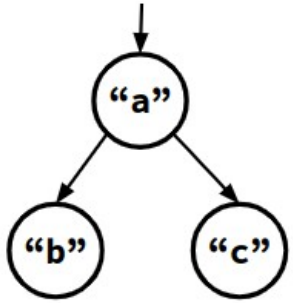
3

?

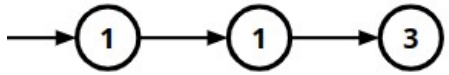
4

?

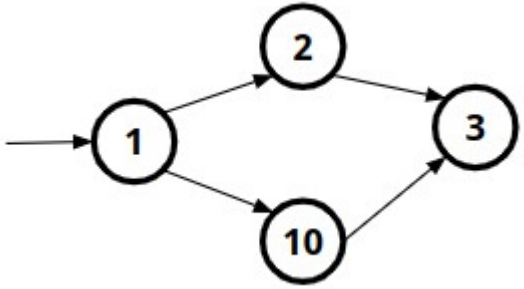
Game: É arvore ou não é?



True ✔



True ✔

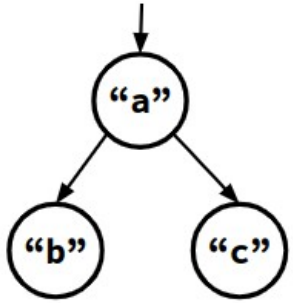


?

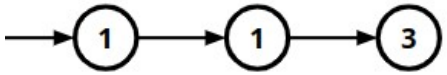
4

?

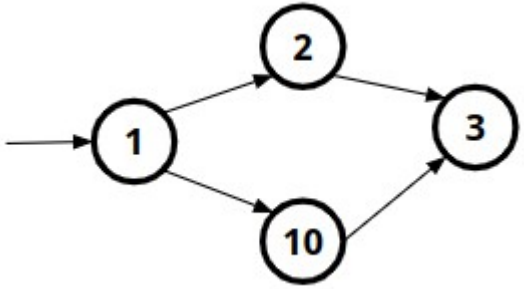
Game: É arvore ou não é?



True ✔



True ✔

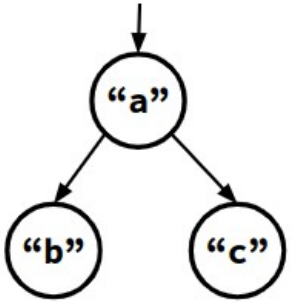


False X

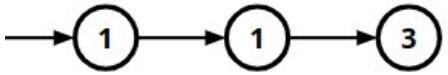
4

?

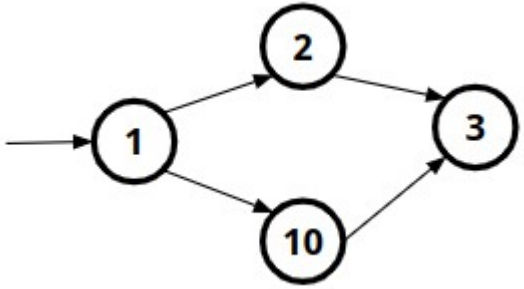
Game: É arvore ou não é?



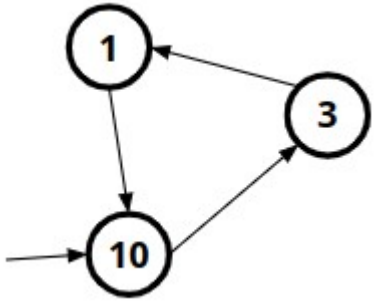
True ✔



True ✔

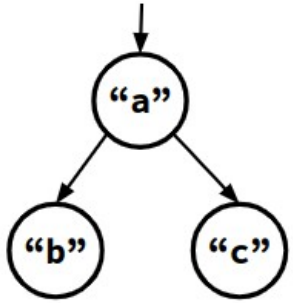


False X

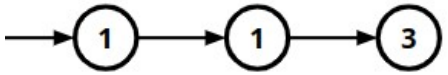


?

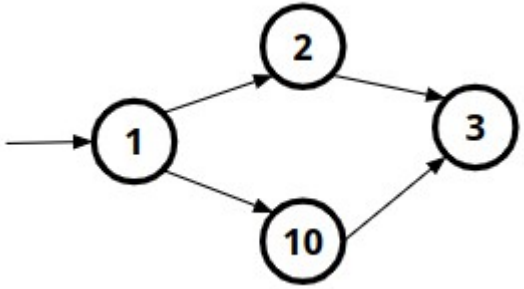
Game: É arvore ou não é?



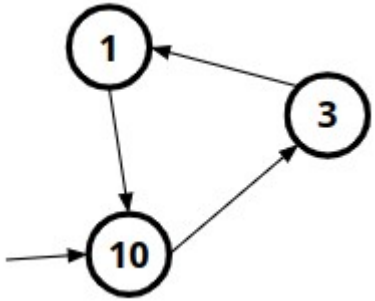
True ✔



True ✔



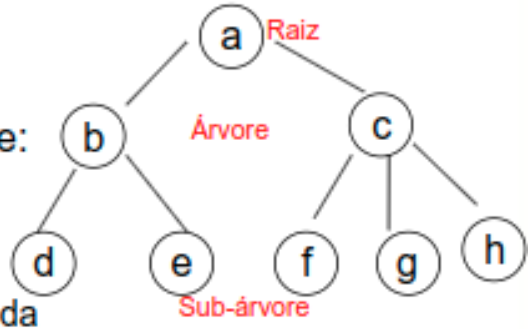
False X



False X

Árvore: Definição

Estruturas de dados não lineares que caracterizam uma relação entre dados, à relação existente entre os dados é uma relação de hierarquia ou de composição.



É um conjunto finito T de um ou mais nós, tais que:

- Existe um só nó principal – RAIZ **(root);**
- Os demais nós formam conjuntos disjuntos onde cada um destes subconjuntos é uma árvore. As arvores recebem a denominação de sub-árvores.

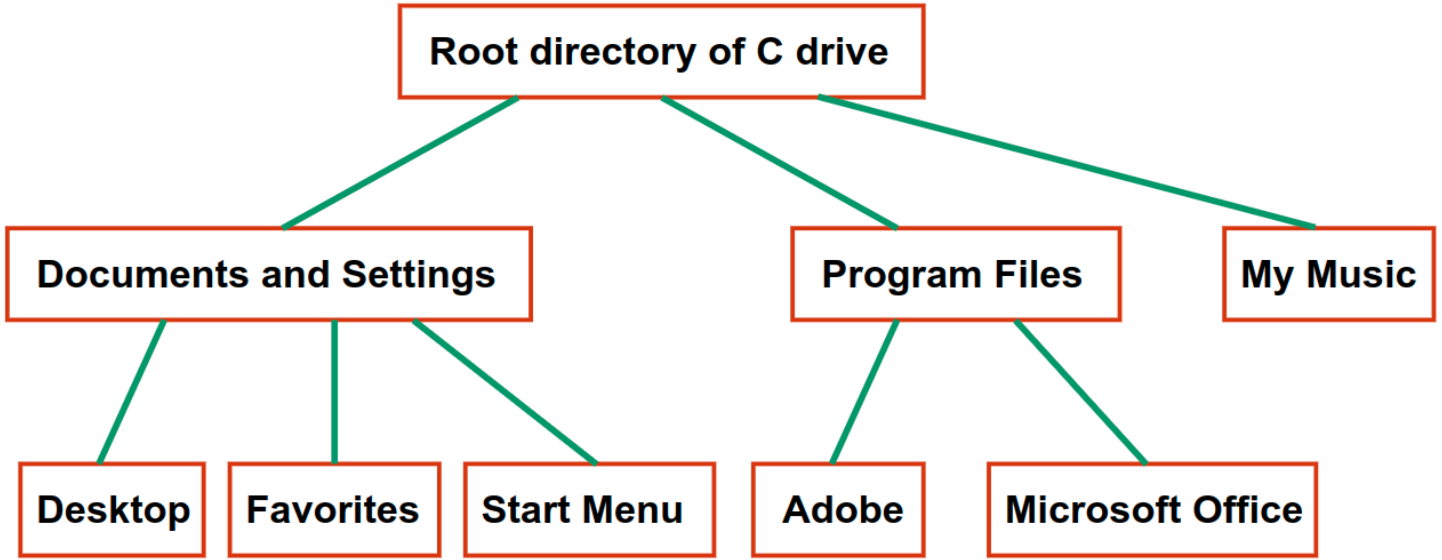
Livro: Art of Computer Programming - Donald Knuth

Árvores

- Uma **árvore** é uma **estrutura de dados não linear** usada para representar entidades que estão em algum relacionamento hierárquico
- **Exemplos** na vida real:
  - Árvore genealógica
  - Índice de um livro
  - Hierarquia de herança de classe em Java
  - Sistema de arquivos de computador (pastas e subpastas)
  - Árvores de decisão
  - Design Top-down

Árvores : Exemplos

- Sistema de arquivos de um computador

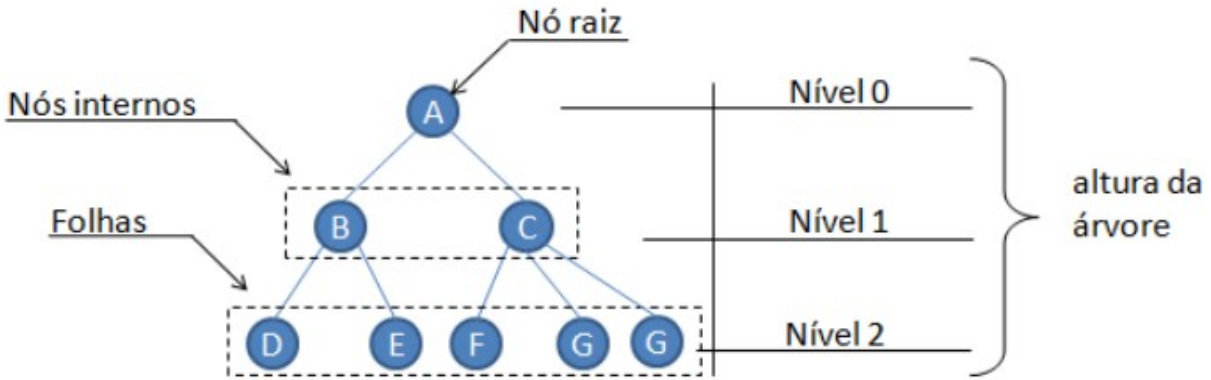


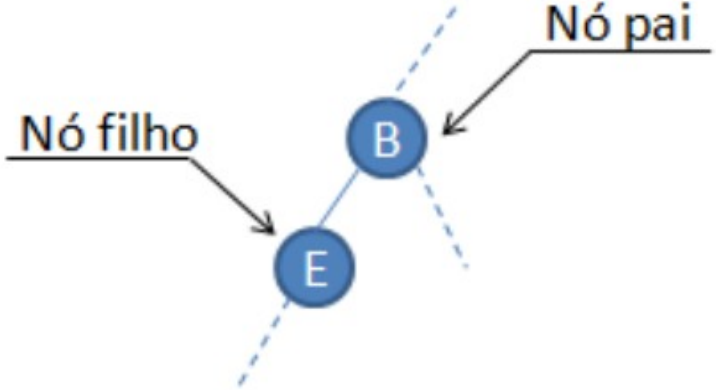
Árvores : Definição

- **Árvore**: um conjunto de elementos do mesmo tipo tal que $\{\}$ ou $\emptyset$
  - É vazio
  - Ou tem um elemento distinto chamado **raiz** da qual descem zero ou mais **árvores** (**subárvores**)

- Que tipo de definição é essa?
  - Qual é o caso básico?
  - Qual é a parte recursiva?

Árvores : Terminologia





Árvores : Terminologia (1/2)

- **Nós**: os elementos na árvore
- **Arestas**: conexões entre nós
- **Raiz**: o elemento distinto que é a origem da árvore
  - Existe apenas um nó raiz em uma árvore
- **Nó folha**: um nó sem aresta para outro nó
- **Nó interno**: um nó que não é um nó folha
- **Árvore vazia:** não tem nós nem arestas

Árvores : Terminologia (2/2)

- **Pai** ou **predecessor:** o nó diretamente acima na hierarquia
  - Um nó pode ter apenas um pai
- **Filho** ou **sucessor**: um nó diretamente abaixo na hierarquia
- **Irmãos**: nós que têm o mesmo pai
- **Ancestrais** de um nó: seu pai, o pai de seu pai, etc.
- **Descendentes** de um nó: seus filhos, os filhos de seus filhos, etc.

Discussão

- Um nó folha tem filhos?
- O nó raiz tem um pai?
- Quantos pais tem cada nó, exceto o nó raiz?

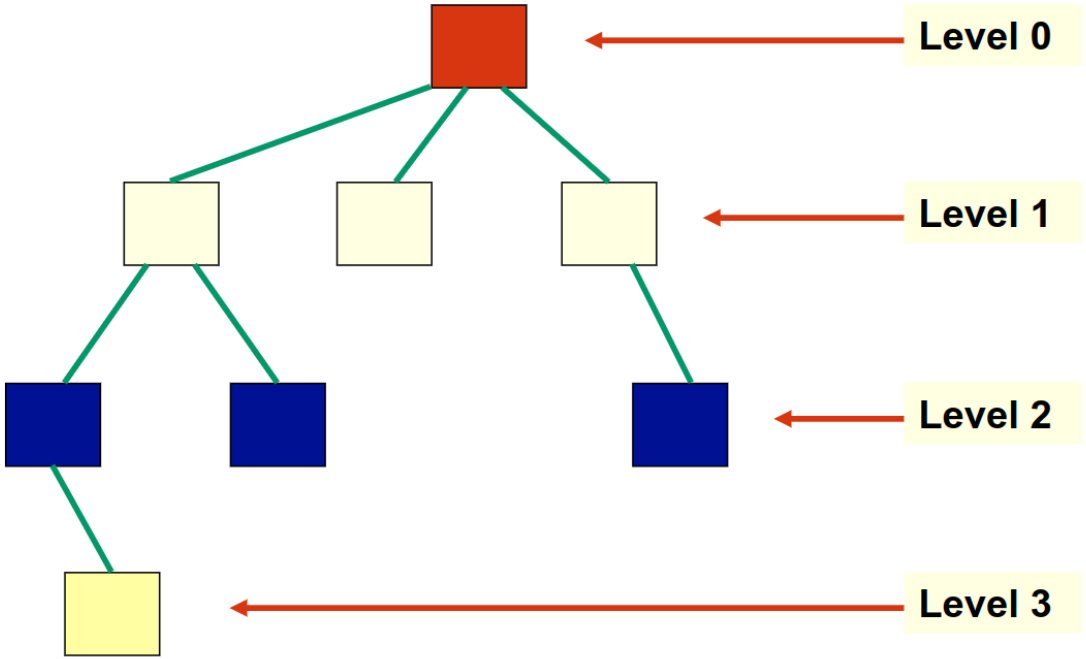
Altura da Árvore

- A informação sobre a **altura da árvore** é um dos principais parâmetros que se devem analisar sobre a estrutura de dados árvore.

Altura da Árvore

- Um **caminho** é uma sequência de arestas que vão de um nó a outro
- **Comprimento de um caminho**: número de arestas no caminho
- **Altura de uma árvore** (não vazia): comprimento do caminho mais longo da raiz até uma folha
  - Qual é a altura de uma árvore que tem apenas um nó raiz?
  - Por convenção, a altura de uma árvore vazia é -1

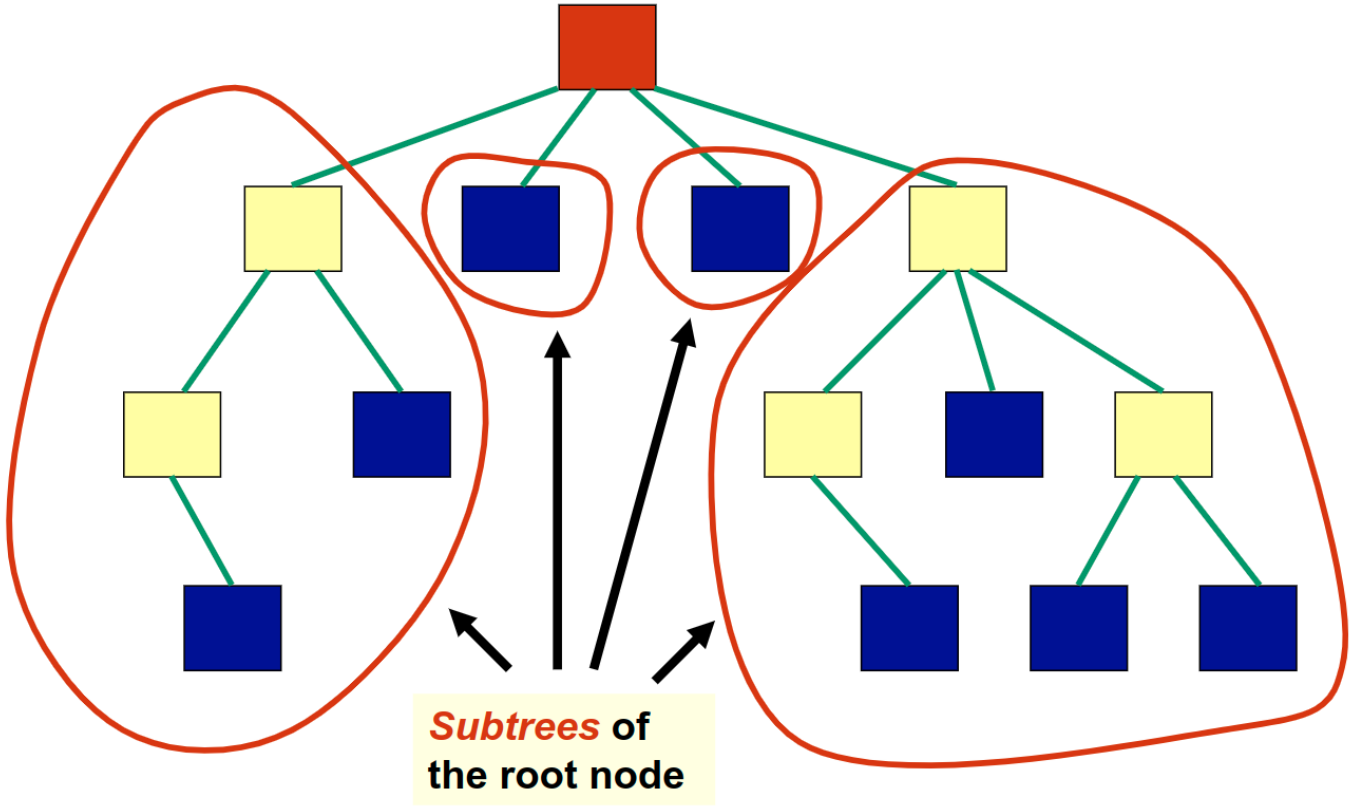
Level de um Nó



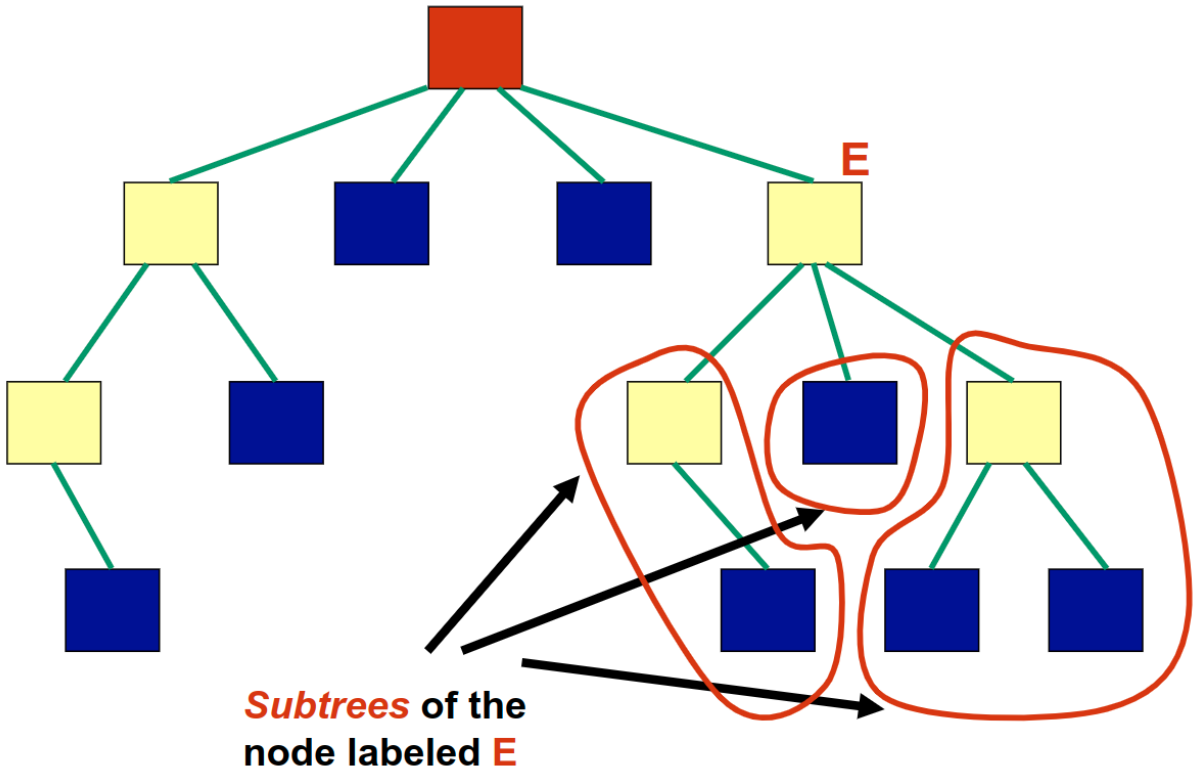
Subárvores

- **Subárvore** de um nó: consiste em um nó filho e todos os seus descendentes
  - Uma **subárvore é** ela mesma **uma árvore**
  - Um nó pode ter muitas subárvores

Subárvores



Subárvores

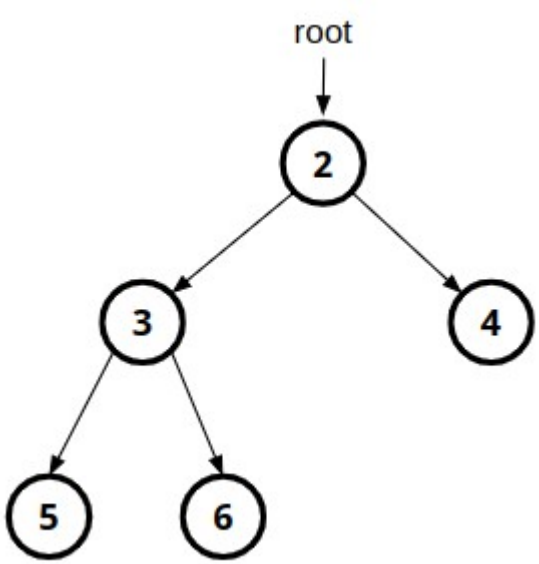


Árvores: Mais terminologia

- **Grau** ou aridade de um **nó**: o número de filhos que possui
- **Grau** ou aridade de uma **árvore**: o máximo dos graus dos nós da árvore

Praticando os conceitos

```
struct Node {
        int data;
        Node* left;
        Node* right;
}
```



Problema: Escreva uma função que recebe a raiz de uma árvore retorne a soma de todas as chaves desta árvore.

find_sum(root) → 20

Este problema deve ser resolvido em O(n)

Árvores Binárias