
4ª EDIÇÃO
Estruturas de Dados
Árvores
Luiza
<CODE>
luizalabs
MAGALU

Ementa:

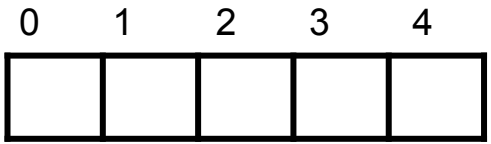
1. Introdução
    a. O que sabemos até aqui
    b. Como escolher qual Estrutura de Dados utilizar?
2. Árvore
    a. Definição e conceitos básicos
    b. Aplicações
3. Árvore binária
    a. Tipos de árvore binária
    b. Propriedades
    c. Implementação
    d. Percursos
4. Árvore de busca binária
    a. Definição
    b. Propriedades
    c. Visualização
5. Exercícios

1. Introdução

1. Introdução

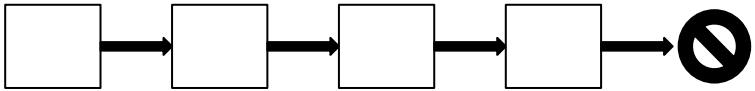
1.1 O que sabemos até aqui?

Até agora aprendemos sobre estruturas de dados elementares como:

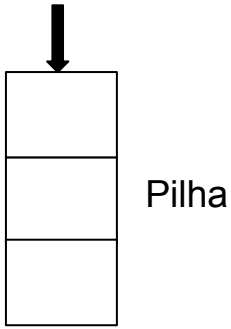


Array

Fila



Lista

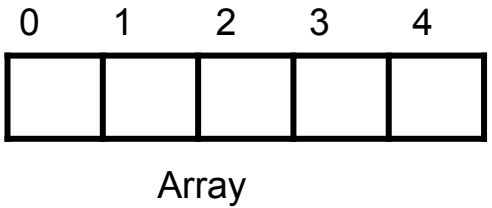


Pilha

1. Introdução

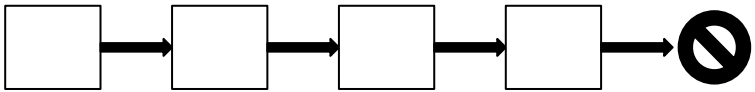
1.1 O que sabemos até aqui?

Nessas estruturas, os dados são estruturados de forma **linear** ou **sequencial**.

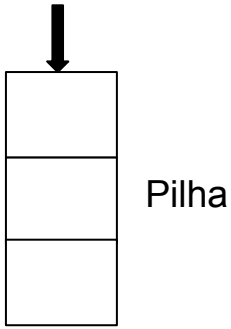


Array

Fila



Lista

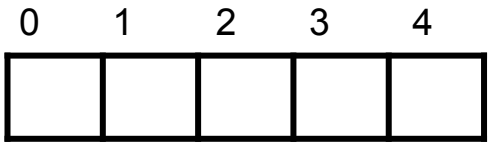


Pilha

1. Introdução

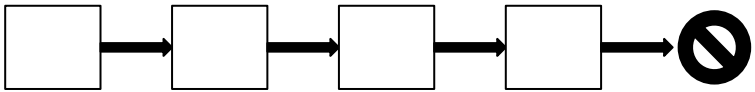
1.1 O que sabemos até aqui?

Nelas há um **início** e **fim** e cada elemento nessas coleções pode ter um **próximo elemento** ou um **elemento anterior.**

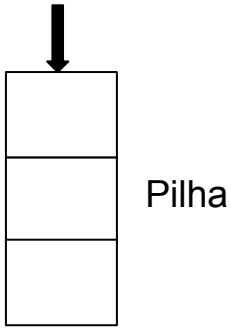


Array

Fila



Lista



Pilha

1. Introdução

1.2 Como escolher qual Estrutura de Dados utilizar?

1. Introdução

1.2 Como escolher qual Estrutura de Dados utilizar?

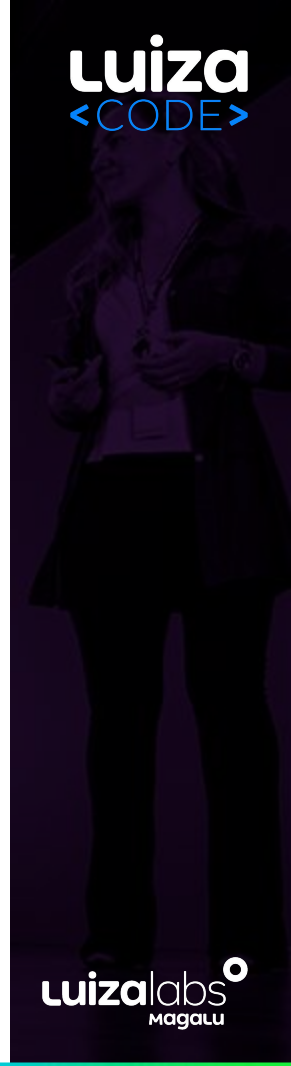
O que vai ser armazenado?

1. Introdução

1.2 Como escolher qual Estrutura de Dados utilizar?

O que vai ser armazenado?

Dependendo da natureza dos dados que vamos guardar, uma estrutura pode ser mais adequada que outras.



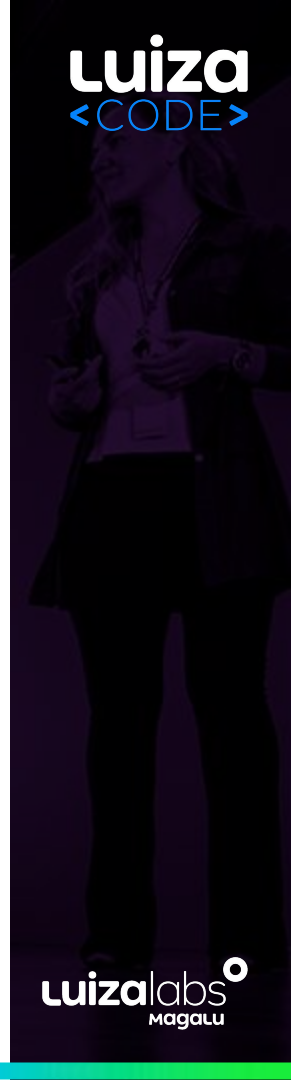
1. Introdução

1.2 Como escolher qual Estrutura de Dados utilizar?

O que vai ser armazenado?

Dependendo da natureza dos dados que vamos guardar, uma estrutura pode ser mais adequada que outras.

Além disso, como esperamos manipular e operar sobre esses dados têm papel determinante sobre a nossa decisão, já que o custo de determinadas operações pode variar entre as estruturas.



1. Introdução

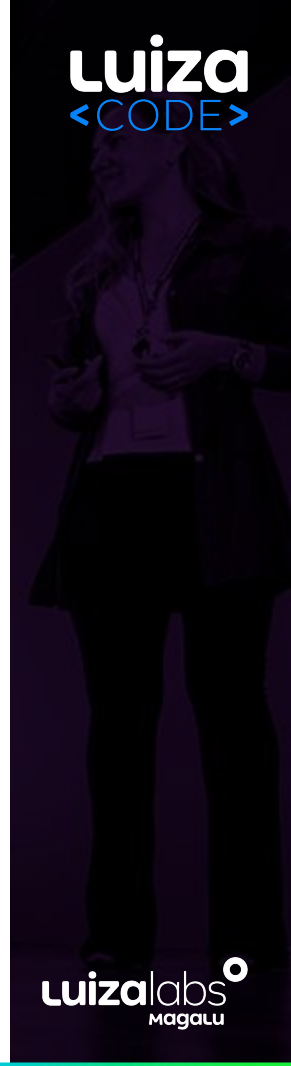
1.2 Como escolher qual Estrutura de Dados utilizar?

O que vai ser armazenado?

Dependendo da natureza dos dados que vamos guardar, uma estrutura pode ser mais adequada que outras.

Além disso, como esperamos manipular e operar sobre esses dados têm papel determinante sobre a nossa decisão, já que o custo de determinadas operações pode variar entre as estruturas.

Mais outro fator é o uso de memória de cada estrutura diante do tamanho e da quantidade de dados que queremos armazenar.

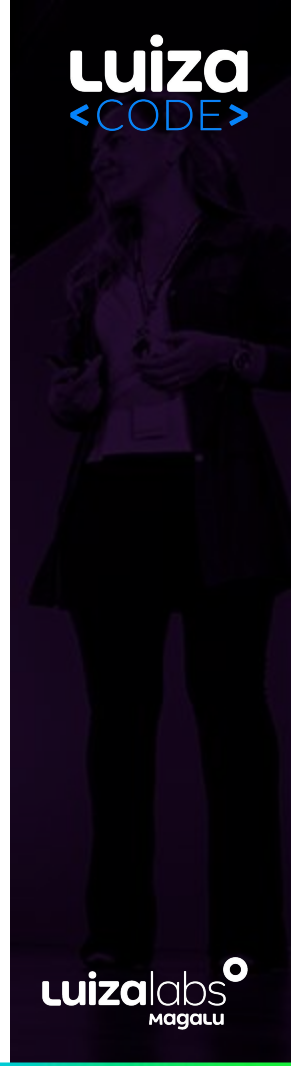


1. Introdução

1.2 Como escolher qual Estrutura de Dados utilizar?

Exemplos:

- Solicitações que devem ser processadas por ordem de chegada

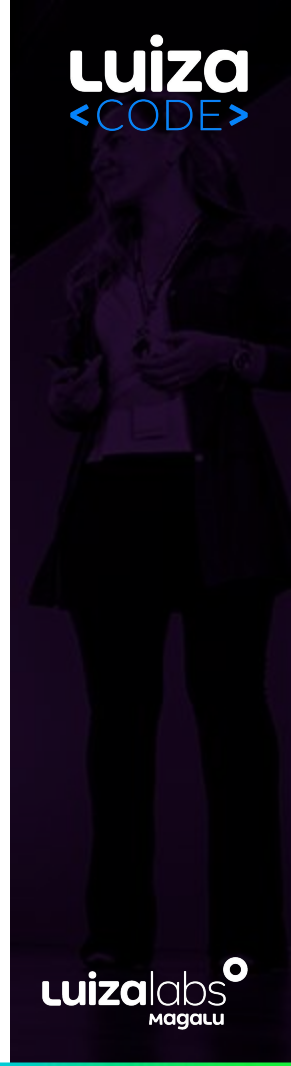


1. Introdução

1.2 Como escolher qual Estrutura de Dados utilizar?

Exemplos:

- Solicitações que devem ser processadas por ordem de chegada
  - **Fila**



1. Introdução

1.2 Como escolher qual Estrutura de Dados utilizar?

Exemplos:

- Solicitações que devem ser processadas por ordem de chegada
  - **Fila**
- Playlist de músicas / vídeos

1. Introdução

1.2 Como escolher qual Estrutura de Dados utilizar?

Exemplos:

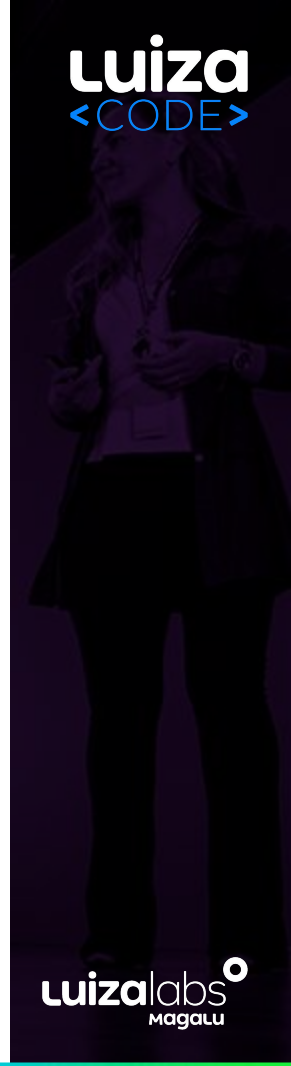
- Solicitações que devem ser processadas por ordem de chegada
  - **Fila**
- Playlist de músicas / vídeos
  - **Fila**

1. Introdução

1.2 Como escolher qual Estrutura de Dados utilizar?

Exemplos:

- Solicitações que devem ser processadas por ordem de chegada
  - **Fila**
- Playlist de músicas / vídeos
  - **Fila**
- Capítulos de um livro

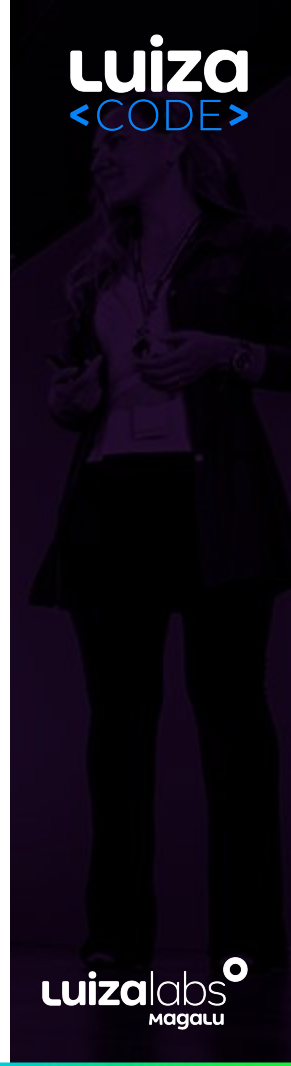


1. Introdução

1.2 Como escolher qual Estrutura de Dados utilizar?

Exemplos:

- Solicitações que devem ser processadas por ordem de chegada
  - **Fila**
- Playlist de músicas / vídeos
  - **Fila**
- Capítulos de um livro
  - **Array**

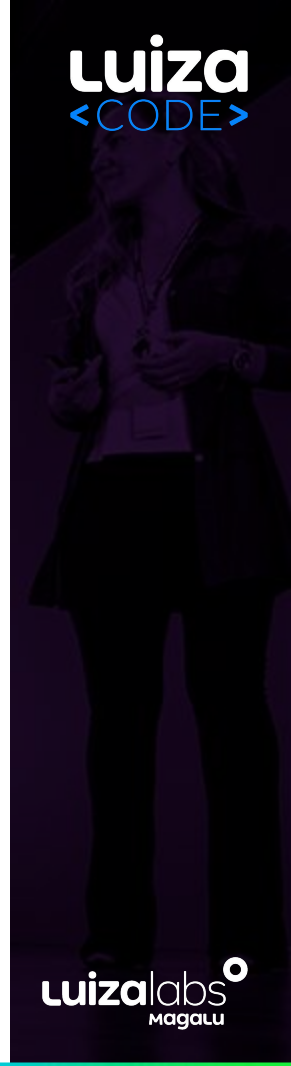



Luiza
<CODE>
2. Árvore
labs
Luizalabs
Magalu

2. Árvore

2.1 Definição

É uma estrutura de dados usada frequentemente para representar dados hierárquicos e decisões / escolhas.



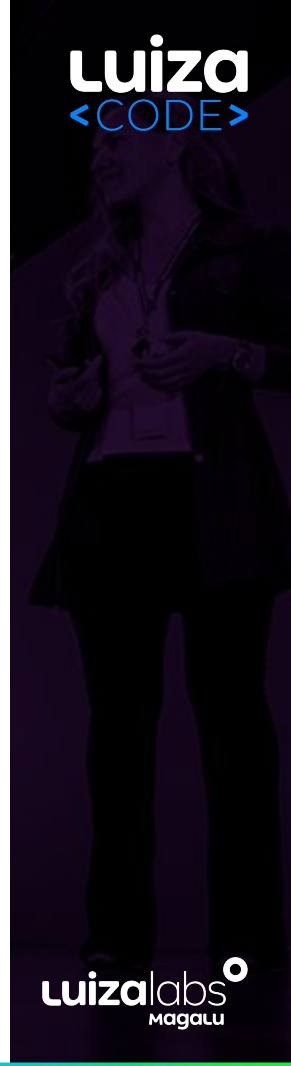
2. Árvore

2.1 Definição

É uma estrutura de dados usada frequentemente para representar dados hierárquicos e decisões / escolhas.

Exemplos:

- Funcionários em uma empresa



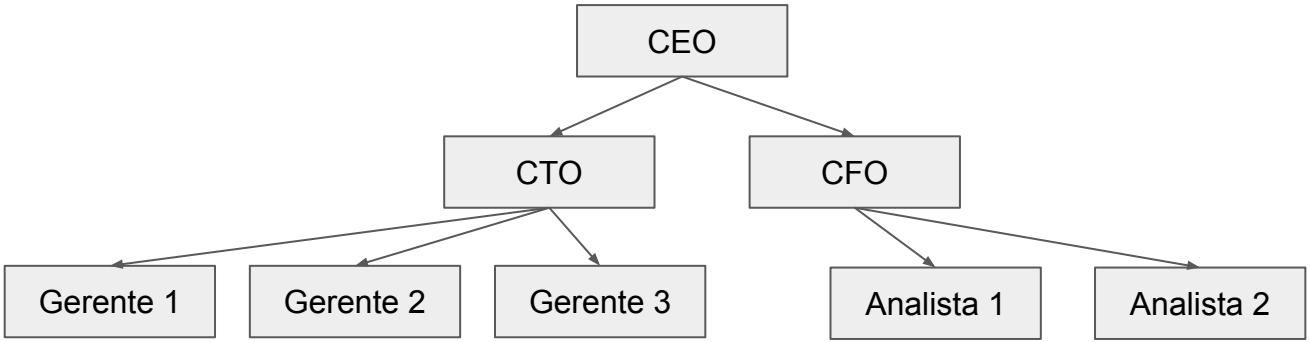
2. Árvore

2.1 Definição

É uma estrutura de dados usada frequentemente para representar dados hierárquicos e decisões / escolhas.

Exemplos:

- Funcionários em uma empresa



2. Árvore

2.1 Definição

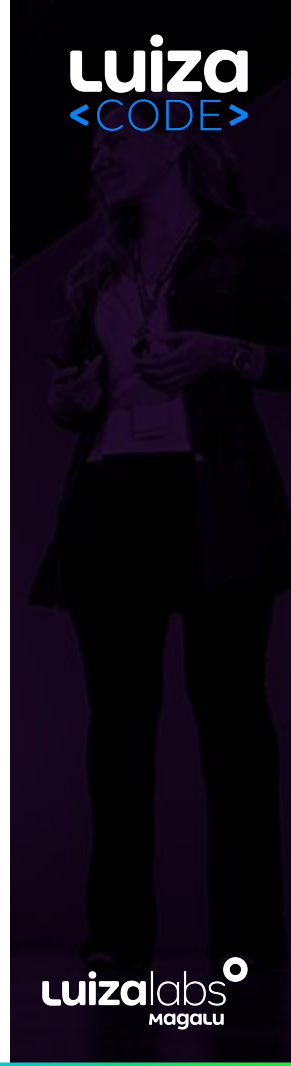
Exemplos:

- Decisões

2. Árvore

2.1 Definição

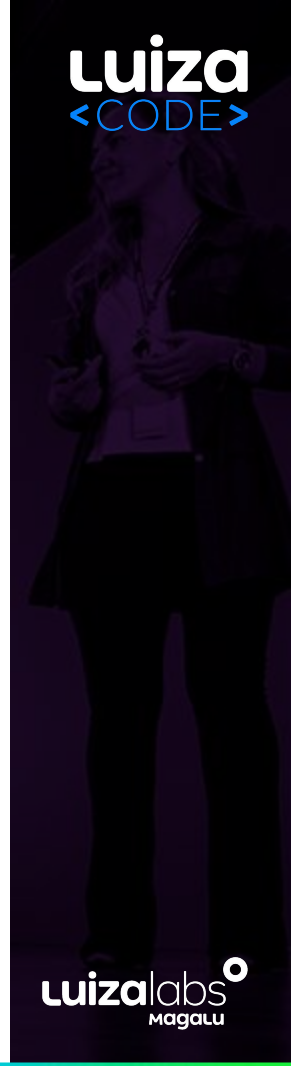
Uma árvore é uma coleção de entidades chamadas **nós**, **conectadas** para simular uma hierarquia.



2. Árvore

2.1 Definição

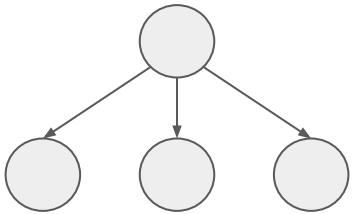
Há um nó ao topo, chamado **raiz**:



2. Árvore

2.1 Definição

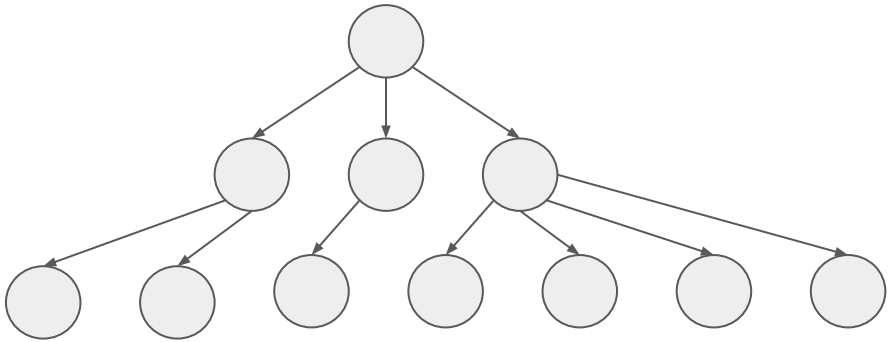
Há um nó ao topo, chamado **raiz**. E a raiz tem referências a **outros nós**:



2. Árvore

2.1 Definição

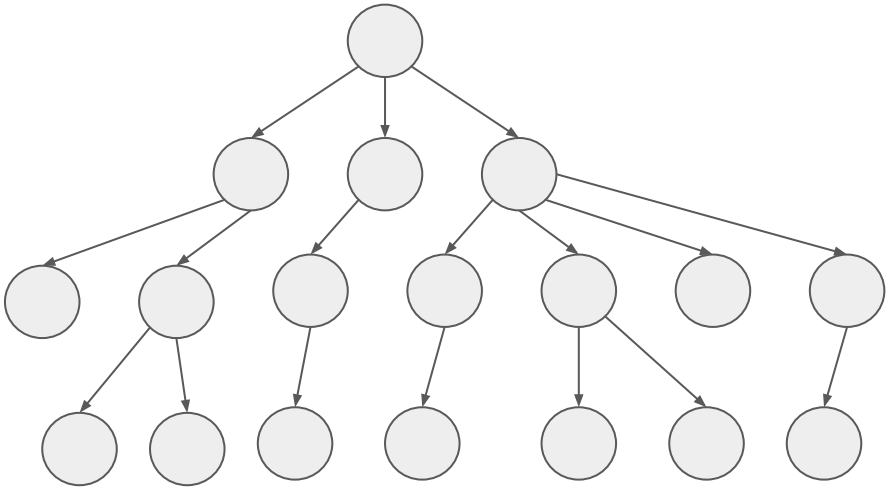
Há um nó ao topo, chamado **raiz**. E a raiz tem referências a **outros nós**. E cada um desses nós pode ter referências a outros nós:



2. Árvore

2.1 Definição

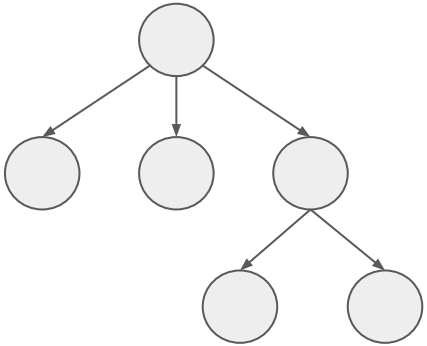
Há um nó ao topo, chamado **raiz**. E a raiz tem referências a **outros nós**. E cada um desses nós pode ter referências a outros nós. E assim sucessivamente:

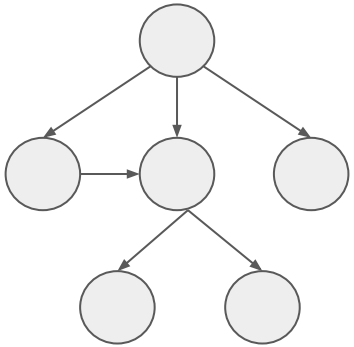


2. Árvore

2.1 Definição

Uma árvore não pode conter **ciclos**:

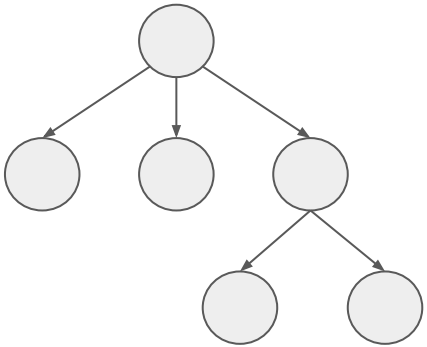


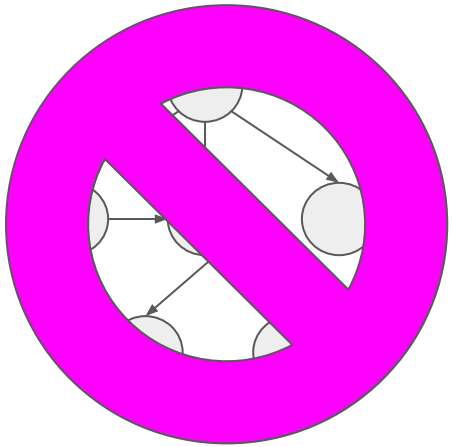


2. Árvore

2.1 Definição

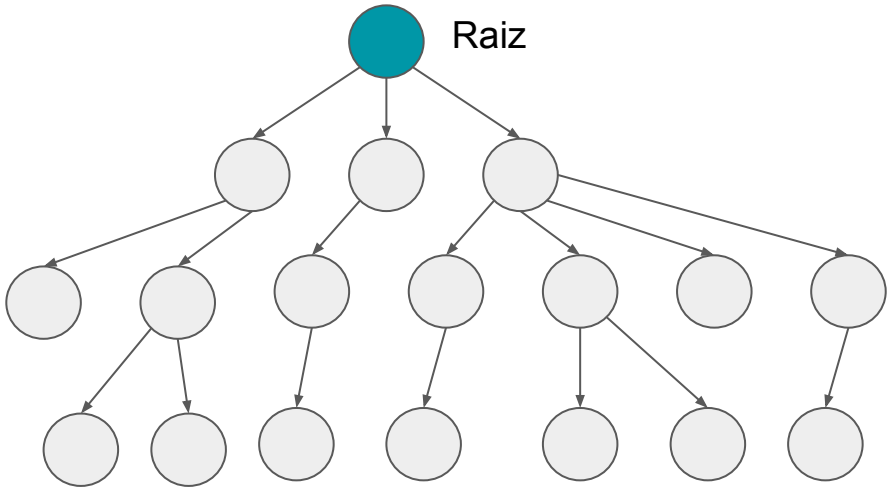
Uma árvore não pode conter **ciclos**:





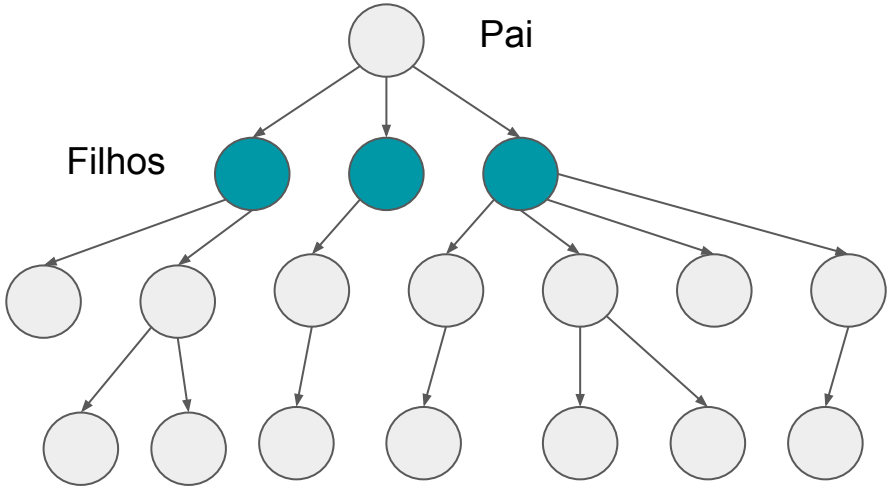
2. Árvore

2.1 Definição



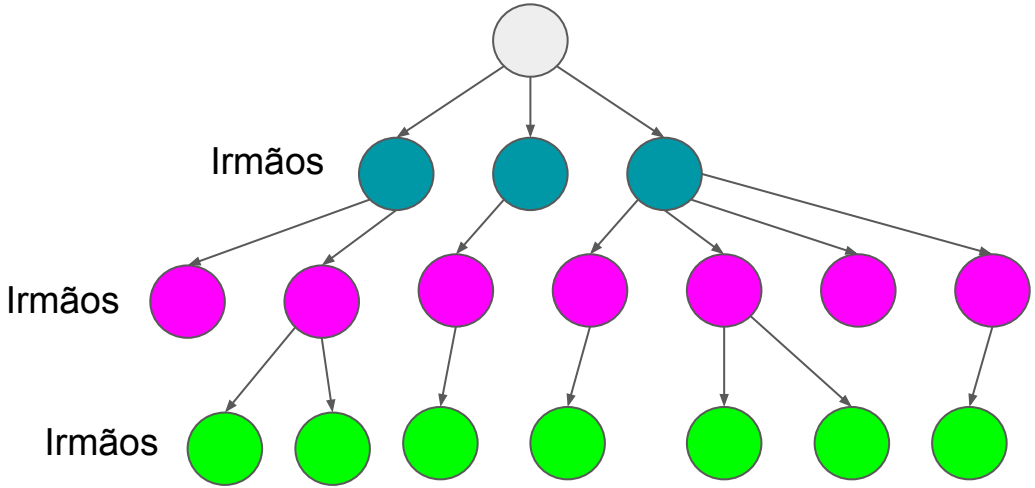
2. Árvore

2.1 Definição



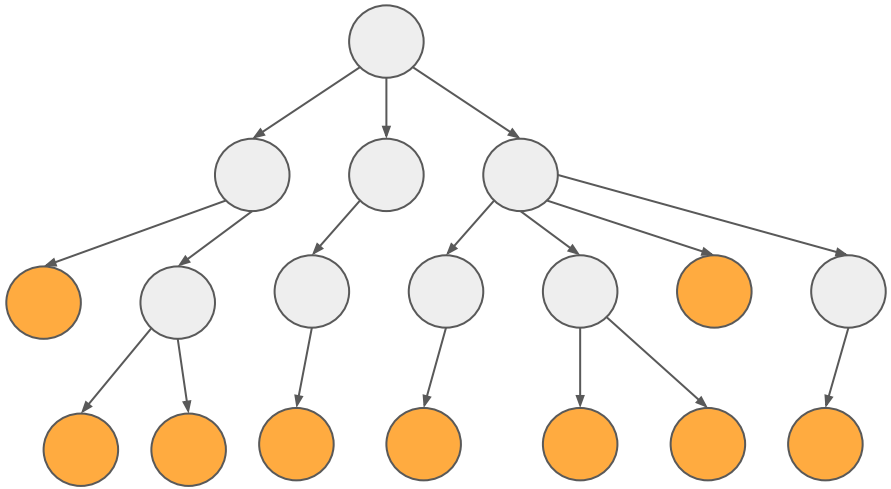
2. Árvore

2.1 Definição



2. Árvore

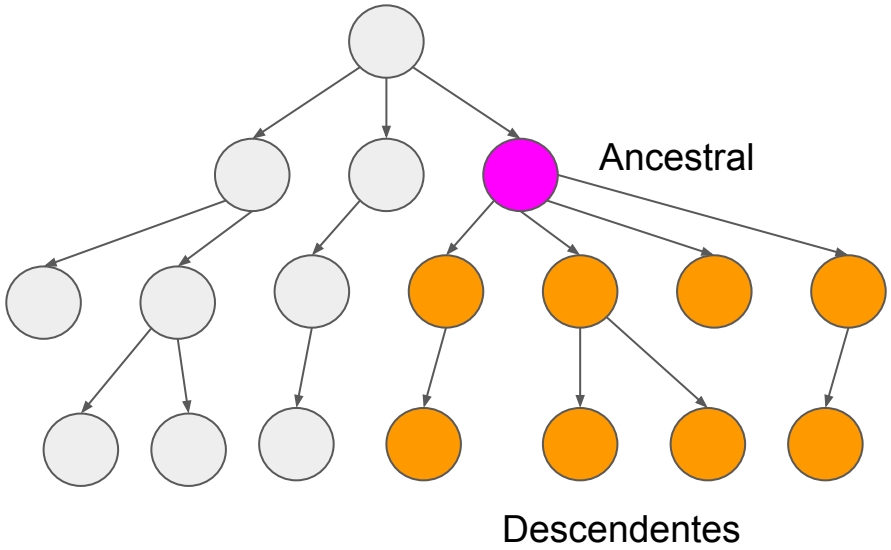
2.1 Definição



Folhas

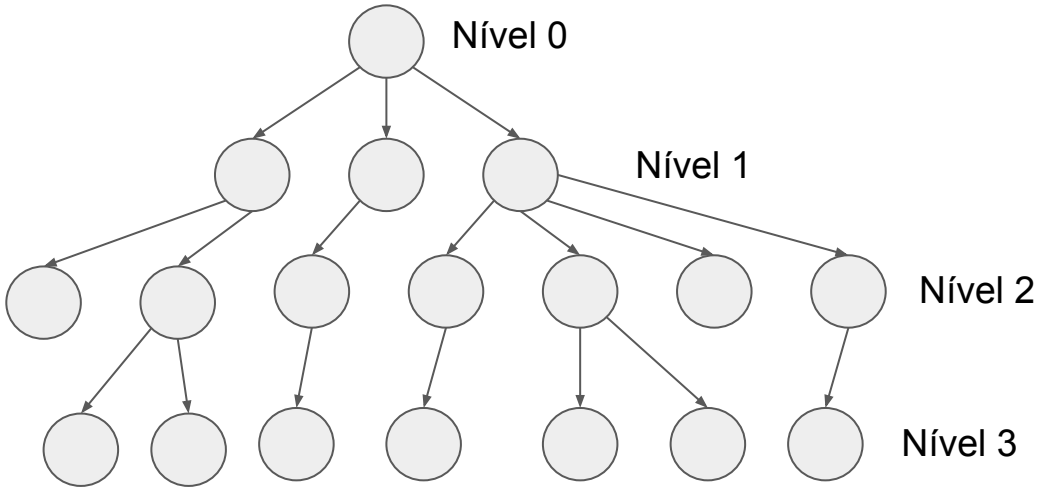
2. Árvore

2.1 Definição



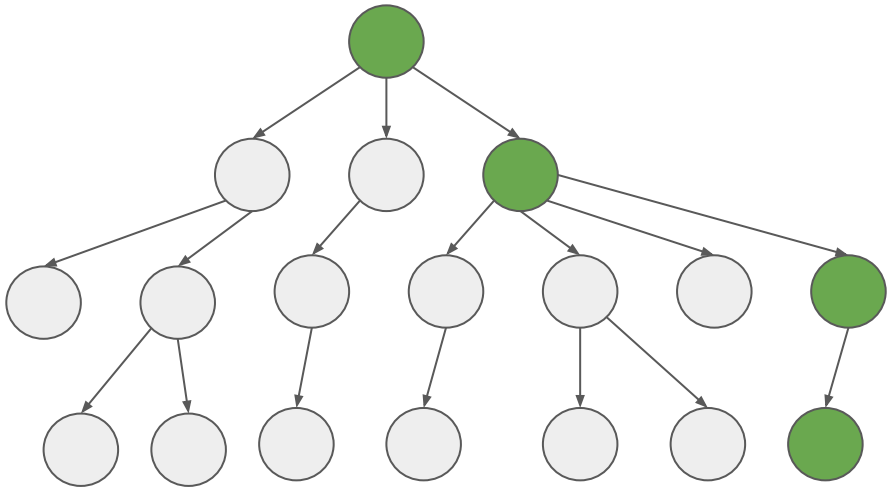
2. Árvore

2.1 Definição



2. Árvore

2.1 Definição

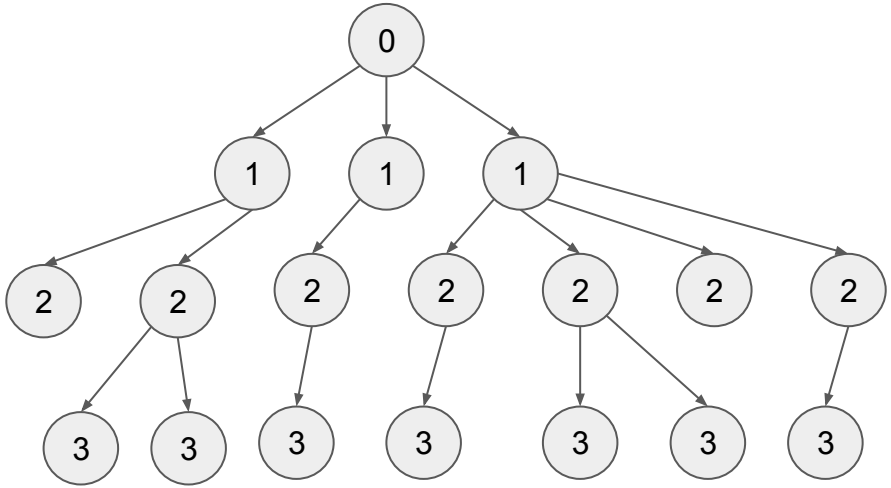


Caminho

2. Árvore

2.1 Definição

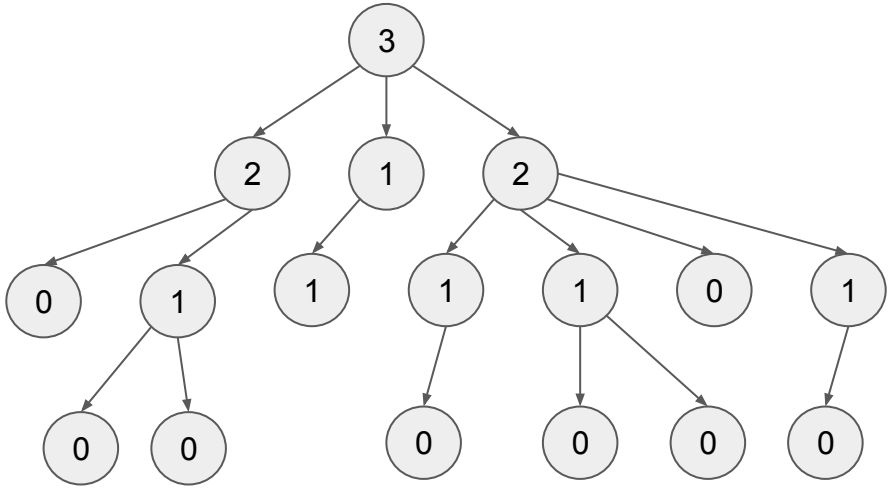
Profundidade de um nó: distância da raiz até o nó.



2. Árvore

2.1 Definição

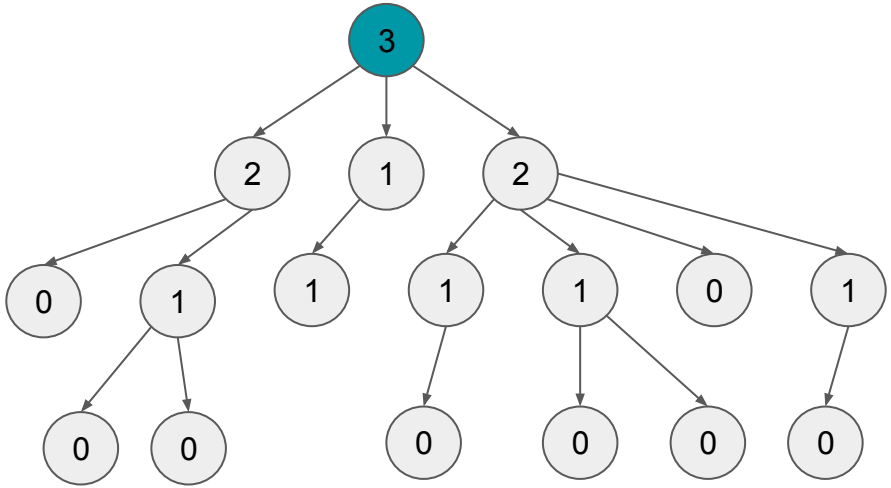
Altura de um nó: distância do nó até a folha mais distante.



2. Árvore

2.1 Definição

Altura da árvore: é a altura da raiz.

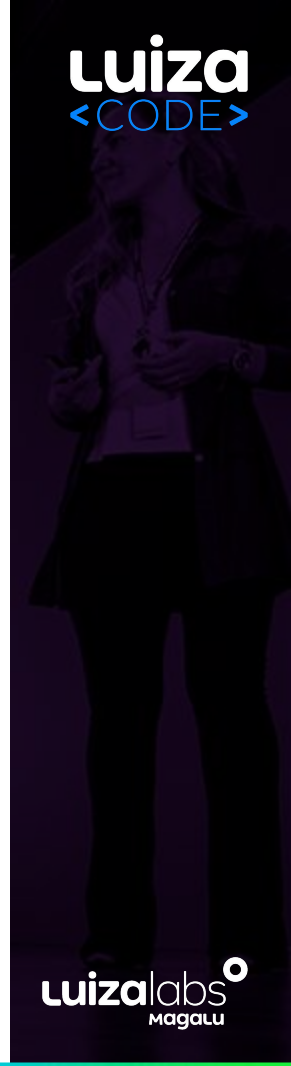


3. Árvore binária

3. Árvore binária

3.1 Definição

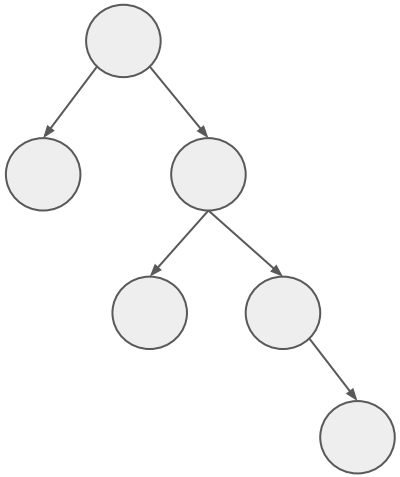
É uma árvore onde cada nó possui **no máximo** 2 filhos.

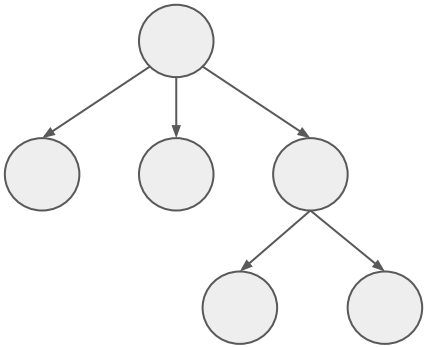


3. Árvore binária

3.1 Definição

É uma árvore onde cada nó possui **no máximo** 2 filhos.

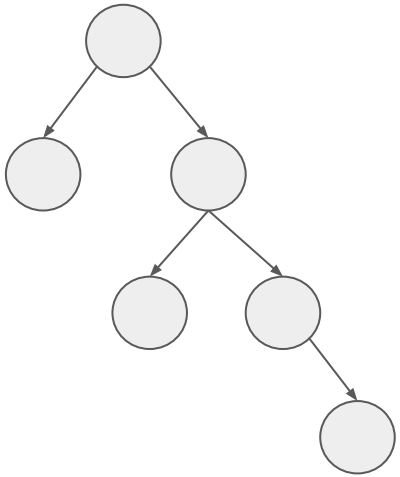


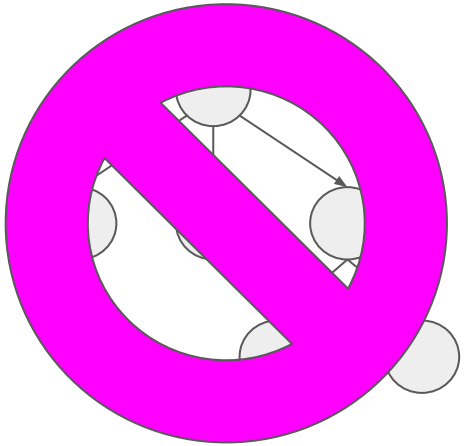


3. Árvore binária

3.1 Definição

É uma árvore onde cada nó possui **no máximo** 2 filhos.



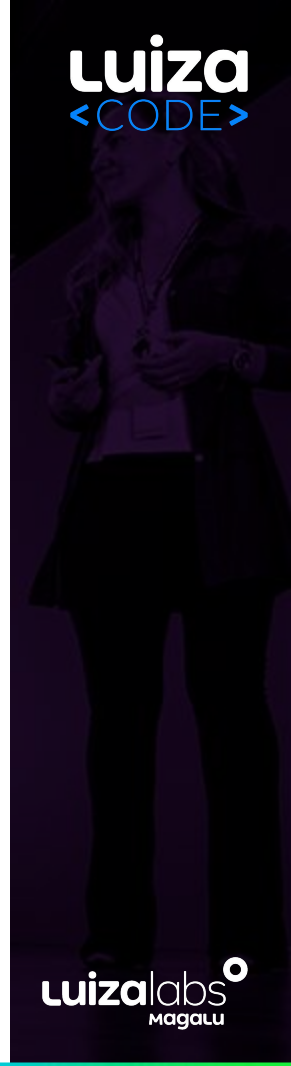


3. Árvore binária

3.2 Tipos de árvore binária

3.2.1 Árvore estritamente binária

É uma árvore binária onde cada nó possui 2 filhos ou nenhum filho.

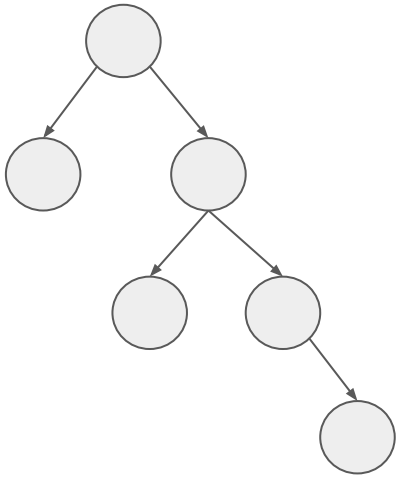


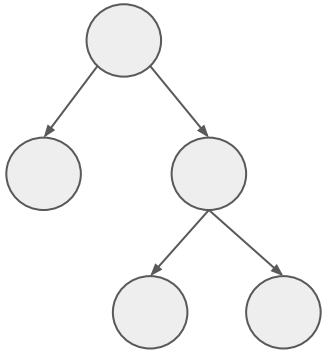
3. Árvore binária

3.2 Tipos de árvore binária

3.2.1 Árvore estritamente binária

É uma árvore binária onde cada nó possui 2 filhos ou nenhum filho.



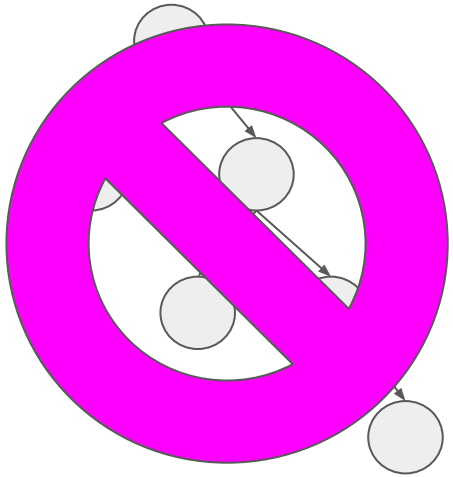


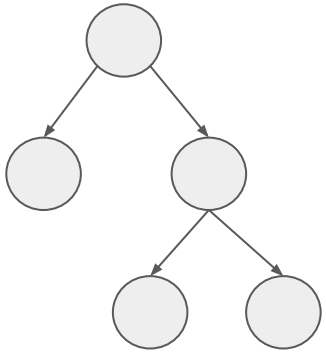
3. Árvore binária

3.2 Tipos de árvore binária

3.2.1 Árvore estritamente binária

É uma árvore binária onde cada nó possui 2 filhos ou nenhum filho.



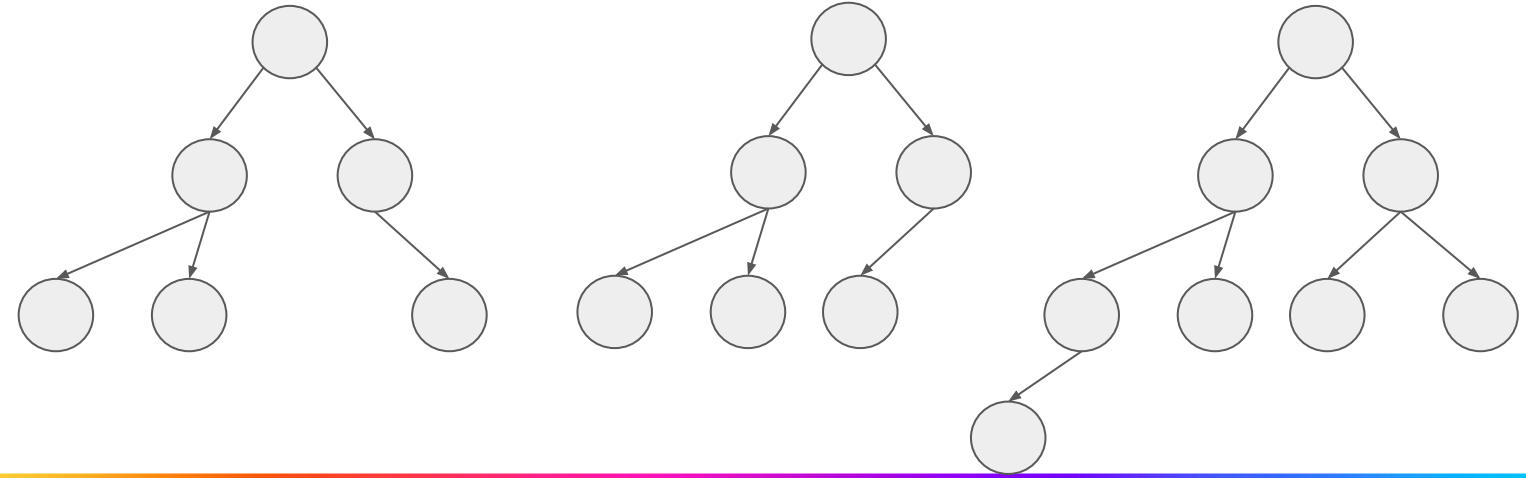


3. Árvore binária

3.2 Tipos de árvore binária

3.2.2 Árvore binária completa

É uma árvore binária onde cada nível está completamente preenchido exceto por, talvez, o último, que é preenchido da esquerda pra direita.

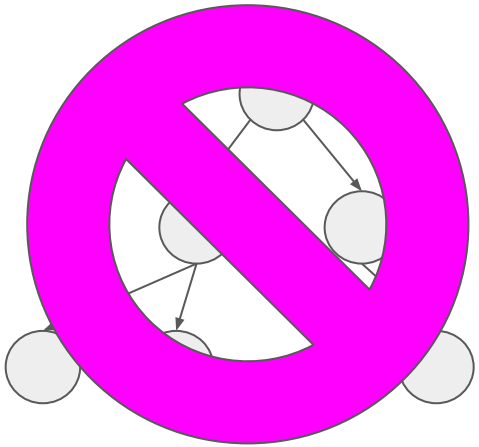


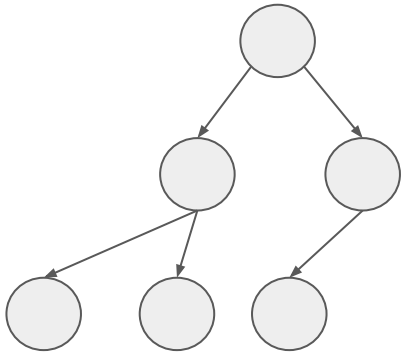
3. Árvore binária

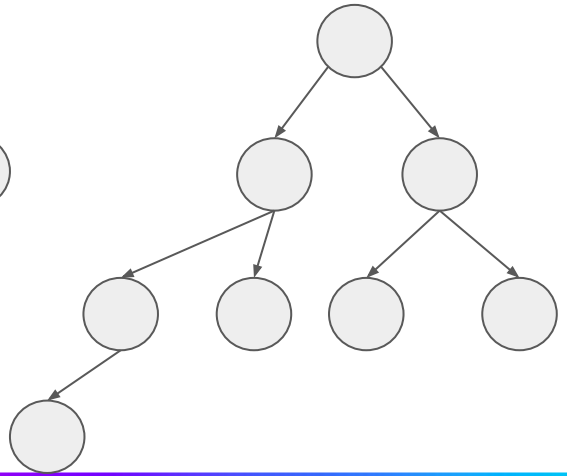
3.2 Tipos de árvore binária

3.2.2 Árvore binária completa

É uma árvore binária onde cada nível está completamente preenchido exceto por, talvez, o último, que é preenchido da esquerda pra direita.





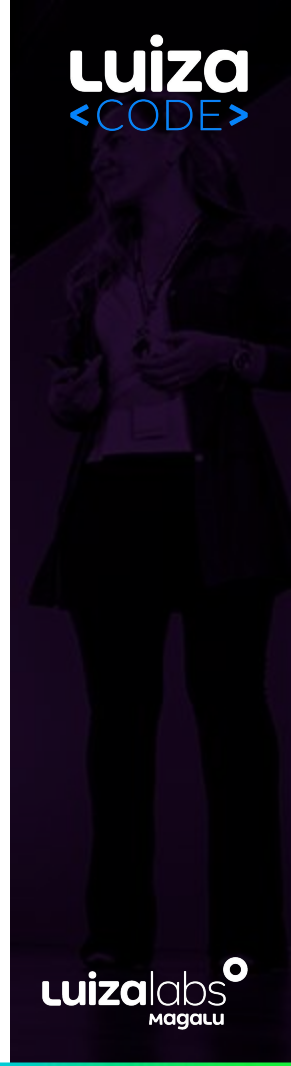


3. Árvore binária

3.2 Tipos de árvore binária

3.2.3 Árvore binária perfeita

É uma árvore binária estritamente binária e completa.

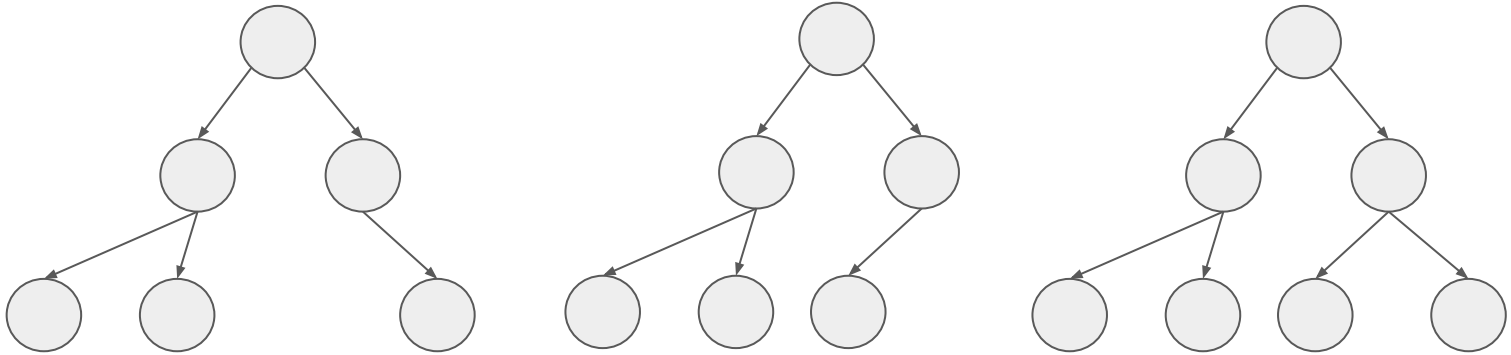


3. Árvore binária

3.2 Tipos de árvore binária

3.2.3 Árvore binária perfeita

É uma árvore binária estritamente binária e completa.

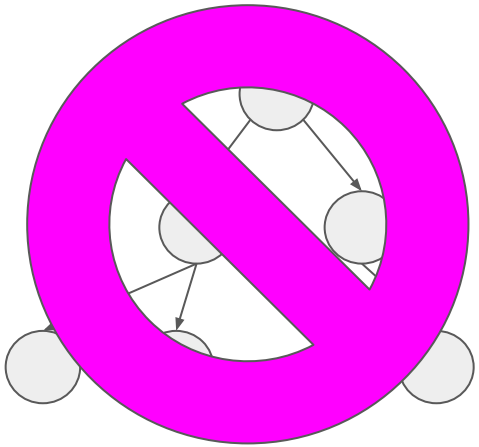


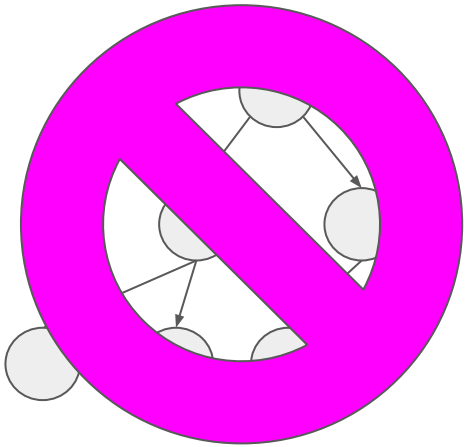
3. Árvore binária

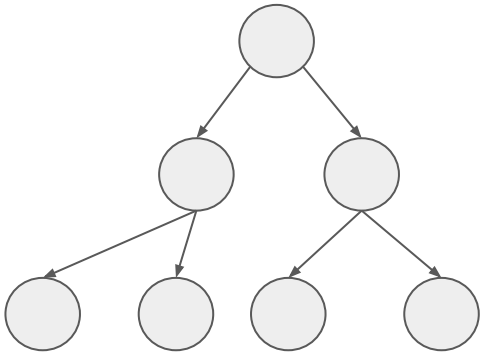
3.2 Tipos de árvore binária

3.2.3 Árvore binária perfeita

É uma árvore binária estritamente binária e completa.





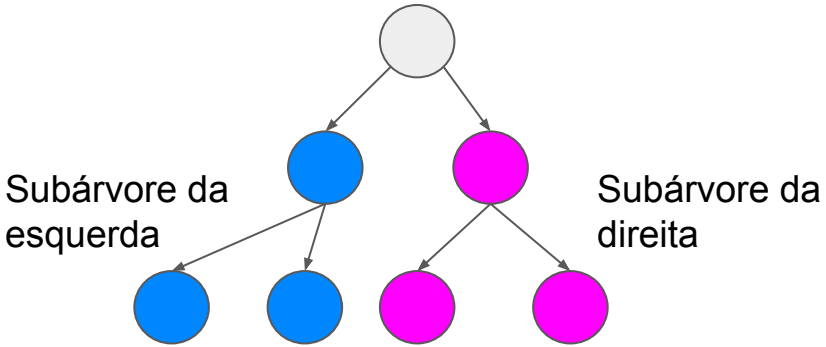


3. Árvore binária

3.2 Tipos de árvore binária

3.2.4 Árvore binária balanceada

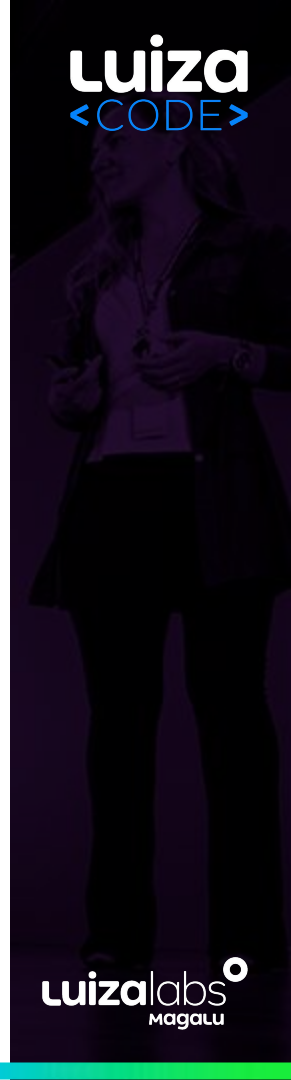
A diferença entre as alturas das subárvores da direita e da esquerda não são maiores que uma determinada constante (geralmente 1).



3. Árvore binária

3.3 Propriedades das árvores binárias

- Número máximo de nós numa árvore binária com altura **h:**
  a. n = 2^0 + 2^1 + 2^2 + ... + 2^h = 2^(h+1) - 1
- Altura de uma árvore binária **perfeita** com **n** nós:
  a. h = log2(n + 1) - 1
- Altura de uma árvore binária **completa** com **n** nós:
  a. h = floor(log2(n + 1))
- Altura máxima de uma árvore binária com **n** nós:
  a. h = n - 1
- Altura mínima de uma árvore binária com **n** nós:
  a. h = floor(log2(n + 1))



3. Árvore binária

3.4 Implementação

Há pelo menos 2 abordagens para implementarmos uma árvore binária:

1. Usando nós alocados dinamicamente com referências aos nós da esquerda e da direita

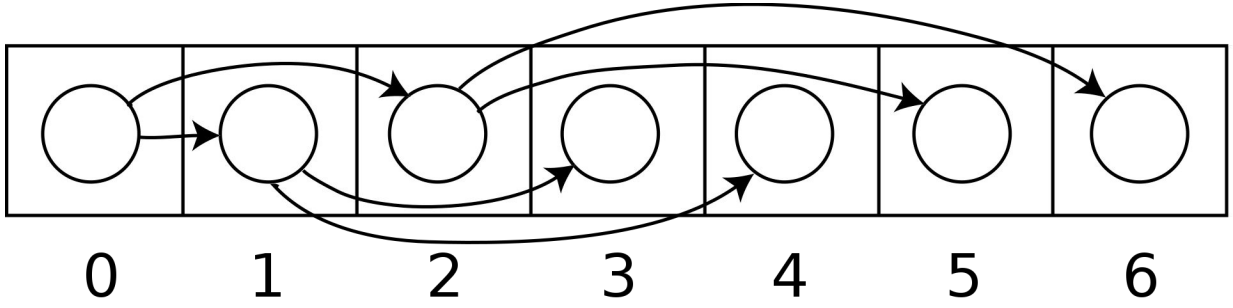
```
class Node {
  constructor(data) {
   this.data = data;
   this.left = null;
   this.right = null;
  }
}

```

3. Árvore binária

3.4 Implementação

Há pelo menos 2 abordagens para implementarmos uma árvore binária:

1. Usando arrays (para uma árvore binária completa):
    a. Para um nó no índice **i** do array
        i. O índice do filho à esquerda desse nó é 2**i** + 1
        ii. O índice do filho à direita desse nó é 2**i** + 2



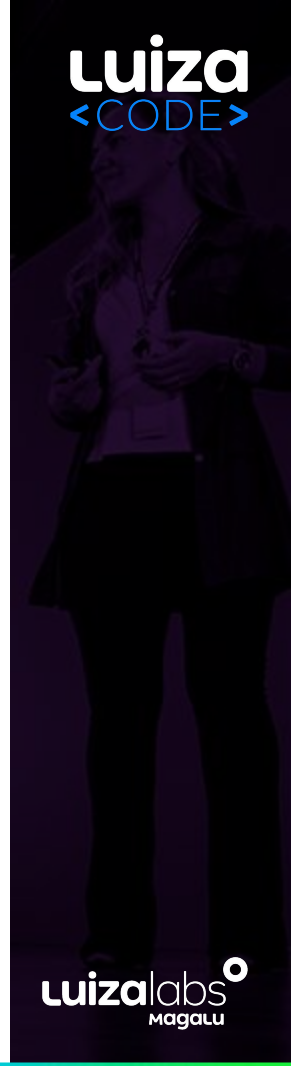
3. Árvore binária

3.5 Percursos

São maneiras de “caminhar” pela árvore e enumerar os seus nós.

Há 3 percursos principais.

- Pré-ordem
- Ordem
- Pós-ordem



3. Árvore binária

3.5 Percursos

3.5.1 Percurso em pré-ordem

Enumera primeiro a raiz antes de percorrer as subárvores da esquerda e da direita.

```
function preorder(root) {
    const nodes = [];
    if (root) {
        nodes.push(root.val);
        preorder(root.left);
        preorder(root.right);
    }
    return nodes;
}
```

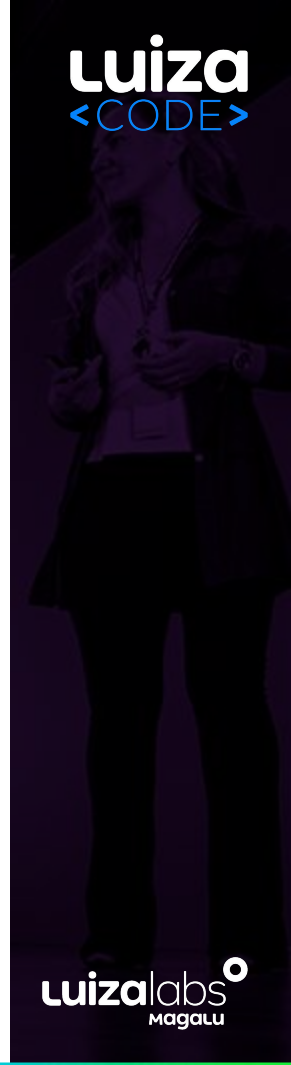
3. Árvore binária

3.5 Percursos

3.5.2 Percurso em ordem

Enumera primeiro a subárvore da esquerda, em seguida, a raiz e, por fim a subárvore da direita.

```
1  const inorder(root) {
2      const nodes = [];
3      if (root) {
4          inorder(root.left);
5          nodes.push(root.val);
6          inorder(root.right);
7      }
8      return nodes;
9  }
```



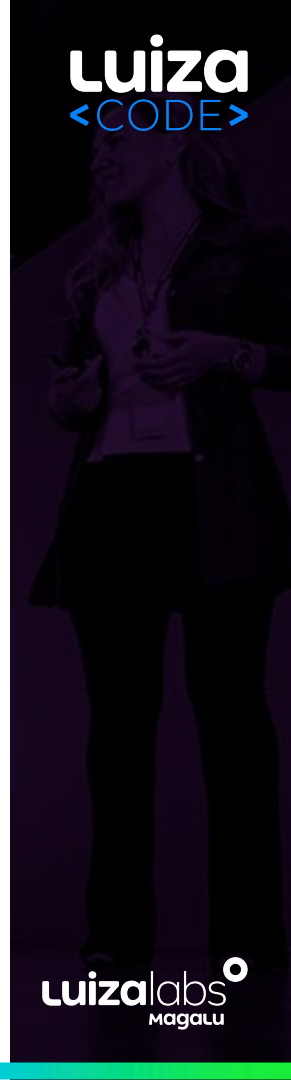
3. Árvore binária

3.5 Percursos

3.5.2 Percurso em pós-ordem

Enumera primeiro a subárvore da esquerda, em seguida, a subárvore da direita e, por fim, a raiz.

```
const postorder(root) {
    const nodes = [];
    if (root) {
        postorder(root.left);
        postorder(root.right);
        nodes.push(root.val);
    }
    return nodes;
}
```



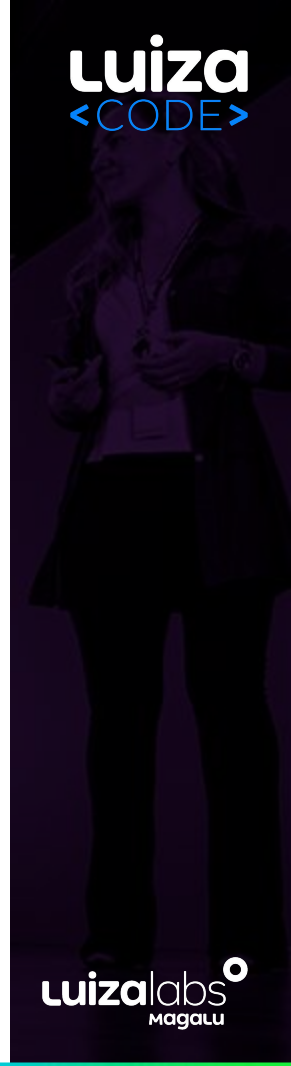
4. Árvore de busca binária

4. Árvore de busca binária

4.1 Definição

É uma árvore binária onde, para cada nó da árvore:

- todos os descendentes à esquerda do nó são **menores ou iguais** a ele
- todos os descendentes à direita do nó são **maiores** que ele.

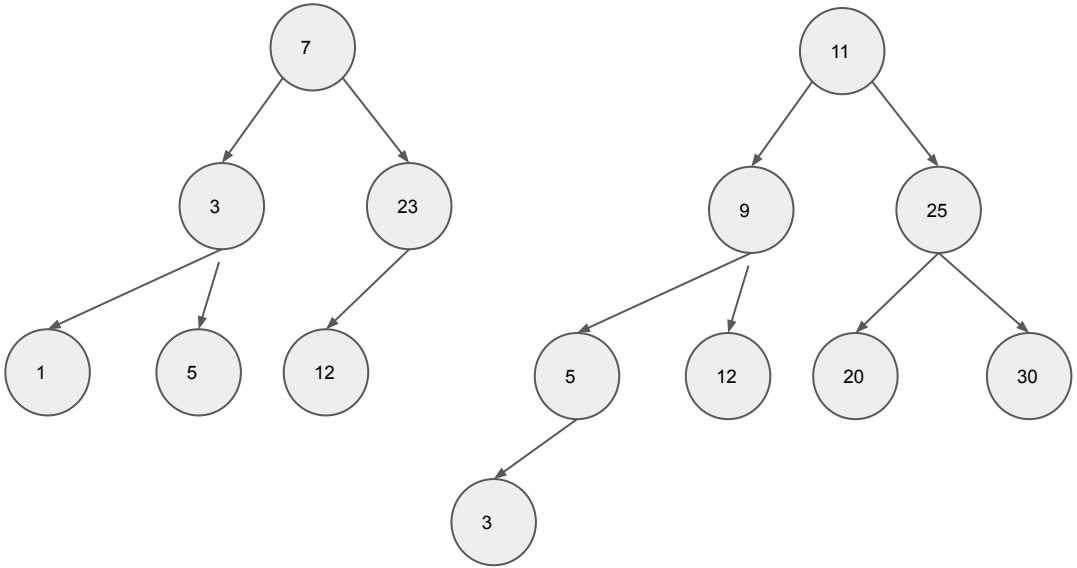


4. Árvore de busca binária

4.1 Definição

É uma árvore binária onde, para cada nó da árvore:

- todos os descendentes à esquerda do nó são **menores ou iguais** a ele
- todos os descendentes à direita do nó são **maiores** que ele.

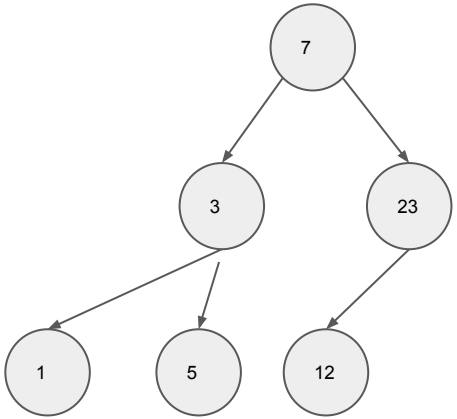


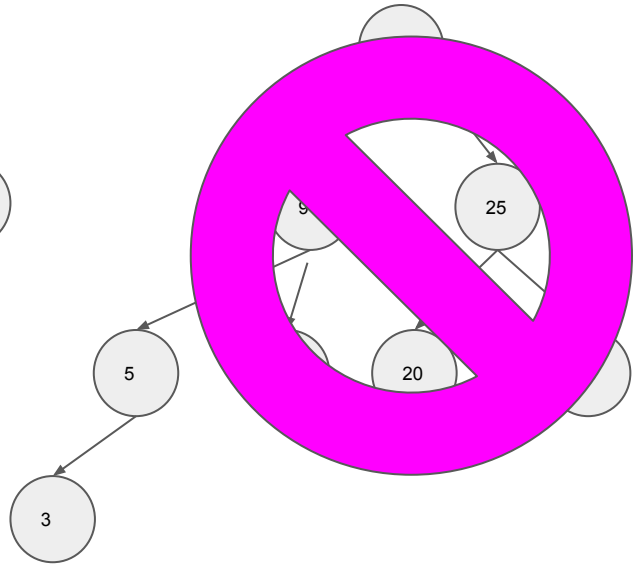
4. Árvore de busca binária

4.1 Definição

É uma árvore binária onde, para cada nó da árvore:

- todos os descendentes à esquerda do nó são **menores ou iguais** a ele
- todos os descendentes à direita do nó são **maiores** que ele.





4. Árvore de busca binária

4.2 Propriedades

- insert - ***O(logn)***

- find - ***O(logn)***

- delete - ***O(logn)***

4. Árvore de busca binária

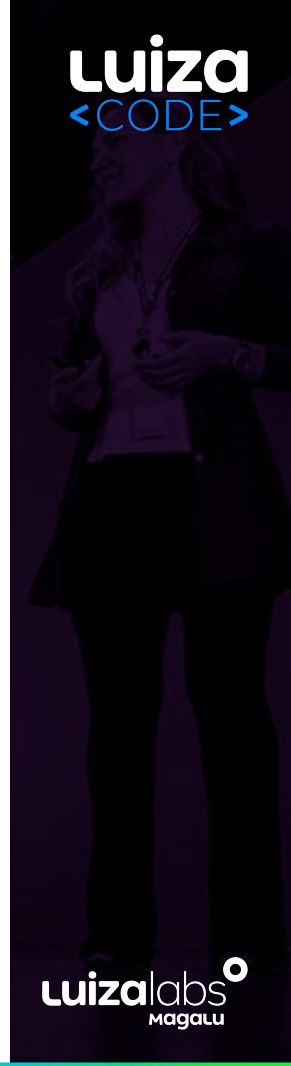
4.3 Visualização

https://www.cs.usfca.edu/~galles/visualization/BST.html

5. Exercícios

5. Exercícios

1. Implemente o cálculo da altura de uma árvore binária
2. Implemente um algoritmo para validar se uma árvore binária é balanceada
3. Implemente um algoritmo para validar de uma árvore binária é uma árvore de busca binária
4. Implemente um algoritmo para encontrar o valor máximo (resp. min) de uma árvore de busca binária
5. Implemente um algoritmo que encontre o sucessor de um nó numa árvore de busca binária



Perguntas?


magalu
#VemSerFeliz