

# Árvores Binárias - Implementação

## Estrutura de Dados

**Professor:** Henrique Viana Oliveira, [henriq.viana@uece.br](mailto:henriq.viana@uece.br)

### Exercício: 1.

Implemente a classe de árvores binárias **LinkedBinaryTree** usando uma estrutura encadeada conforme descrito na Seção 8.3.1.

### Exercício: 2.

Implemente a classe de árvore binária **ArrayListBinaryTree** usando a representação baseada em array descrita na Seção 8.3.2. Implemente as mesmas funções presentes na classe **LinkedBinaryTree**.

### Exercício: 3.

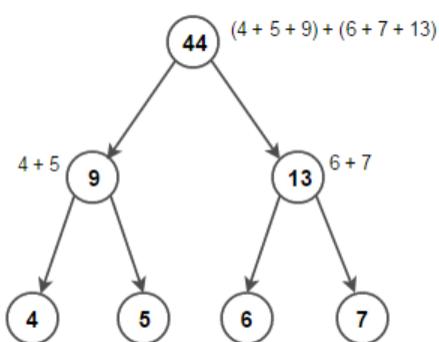
Implemente as funções **Postorder**, **Inorder** e **Preorder Traversal** (seção 8.4.4) para a classe **LinkedBinaryTree**.

### Exercício: 4.

Escreva um algoritmo eficiente para verificar se duas árvores binárias são idênticas ou não. Duas árvores binárias são idênticas se tiverem a mesma estrutura e o mesmo conteúdo.

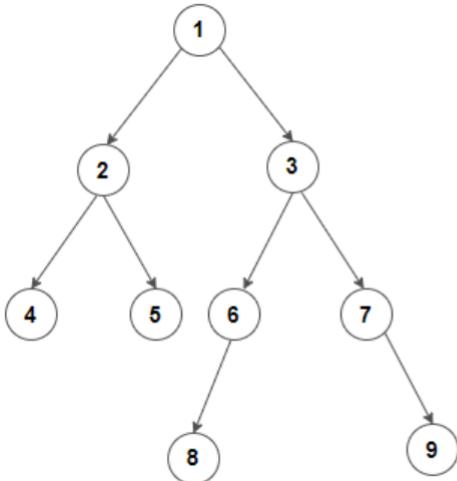
### Exercício: 5.

Dada uma árvore binária, verifique se ela é uma árvore soma ou não. Em uma árvore soma, o valor de cada nó não folha é igual à soma de todos os elementos presentes em suas subárvores esquerda e direita. O valor de um nó folha pode ser qualquer um e o valor de um nó vazio é considerado 0.



**Exercício: 6.**

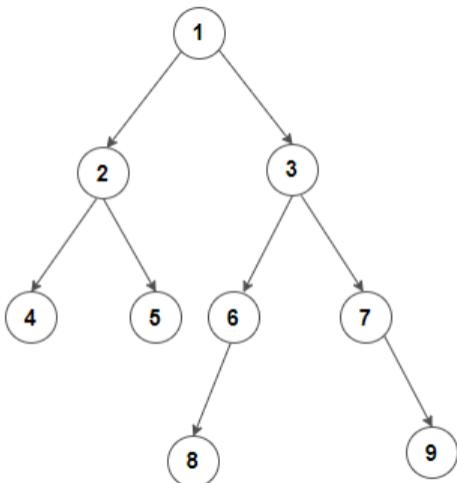
Dada uma árvore binária, escreva um algoritmo eficiente para imprimir todos os caminhos do nó raiz até cada nó folha.



1 → 2 → 4  
1 → 2 → 5  
1 → 3 → 6 → 8  
1 → 3 → 7 → 9

**Exercício: 7.**

Dada uma árvore binária, encontre todos os ancestrais de um nó específico nela.



Os ancestrais do nó 9 são 7, 3 e 1.  
Os ancestrais do nó 6 são 3 e 1.  
Os ancestrais do nó 5 são 2 e 1.

**Exercício: 8.**

Dada uma árvore binária, substitua o valor de cada nó pela soma de todos os elementos presentes em suas subárvore esquerda e direita. Você pode assumir que o valor de um nó vazio é 0.

