

## Exercícios de ED II - BST

### Exercício 1 – Verificação de Similaridade Estrutural entre ABBs

Construa uma classe que represente uma ABB em C++ e implemente um método para determinar se duas árvores binárias de busca são estruturalmente semelhantes, independentemente dos valores armazenados em cada nó.

#### Requisitos:

- Método `bool estruturalmenteSimilar(Arvore &outraArvore)` que retorna `true` se ambas possuírem a mesma estrutura.
- Teste seu método com diferentes exemplos de árvores similares e não similares.

### Exercício 2 – Impressão Visual de uma ABB

Crie uma classe que implemente uma ABB com um método avançado para imprimir visualmente a árvore no console, exibindo claramente sua estrutura hierárquica.

#### Requisitos:

- Implementar um método `void imprimirVisual()` que apresente uma visualização clara, nível por nível, utilizando indentação ou caracteres especiais para representar cada nível.
- A impressão deve ser clara e de fácil interpretação.
- Realize testes inserindo diversos valores para formar árvores diferentes e valide visualmente os resultados.

### Exercício 3 – ABB com Método para Converter em Lista Encadeada Ordenada

Implemente uma classe ABB que possua um método adicional capaz de converter a árvore em uma lista simplesmente encadeada ordenada (em ordem crescente), usando a mesma estrutura de dados original da árvore.

#### Requisitos:

- Crie um método `Node* converterParaListaEncadeada()` que retorne um ponteiro para o primeiro nó da lista encadeada criada (não use nós novos, reorganize apenas os ponteiros da árvore original).
- Exiba o resultado após a conversão, percorrendo a lista gerada.

## Exercício 4 – ABB Indexada (Recuperação por Ordem)

Crie uma classe que implemente uma ABB especial que permita a recuperação de elementos pela sua posição ordenada (indexação por ordem crescente).

### Requisitos:

- Cada nó deve armazenar a quantidade total de nós existentes na sua subárvore (inclusive ele mesmo).
- Implemente um método `int encontrarKthElemento(int k)` que retorna o k-ésimo menor elemento da ABB.
- Ao inserir/remover elementos, atualize corretamente o contador de nós em cada nó afetado.
- Faça testes para verificar o comportamento ao recuperar diferentes posições ordenadas da árvore.