

Relatório Avaliação RA02

Andrey Cabral Meira

Guilherme de Quadro Daudt e João Pedro Fonseca

Árvore Binária

O primeiro código implementa uma Árvore Binária de Busca (BST) em Java. A Árvore Binária de Busca é uma estrutura de dados que organiza elementos de maneira hierárquica, onde cada nó possui no máximo dois filhos: um à esquerda e outro à direita. A estrutura é construída de forma que os elementos menores que um nó é armazenado na subárvore esquerda, enquanto os elementos maiores são armazenados na subárvore direita.

Funcionalidades Principais:

- . Inserção de elementos na árvore.
- . Percorrendo a árvore nas formas pré-ordem, inordem e pós-ordem.
- . Remoção do maior elemento.
- . Remoção do menor elemento.
- . Remoção de um elemento específico.

Uso do Programa:

O programa permite ao usuário escolher entre várias operações, como inserir elementos, percorrer a árvore, remover o maior/menor elemento ou remover um elemento específico. As operações são selecionadas a partir de um menu no console.

Árvore AVL

O segundo código implementa uma Árvore AVL em Java. A Árvore AVL é uma variação das Árvores Binárias de Busca (BST), com a propriedade adicional de estar sempre balanceada. Isso significa que a diferença entre as alturas das subárvores esquerda e direita de cada nó é no máximo 1. Essa propriedade garante que as operações de busca, inserção e remoção tenham complexidade de tempo $O(\log n)$, onde "n" é o número de elementos na árvore.

Funcionalidades Principais:

- . Inserção de elementos na árvore com balanceamento automático.
- . Busca de elementos na árvore.
- . Remoção de elementos com reequilíbrio.

Uso do Programa:

O programa permite ao usuário escolher entre operações como inserir elementos aleatórios com semente, buscar elementos na árvore, remover elementos e sair do programa. Os resultados das operações são apresentados no console.

Conclusão

A escolha entre os códigos depende das necessidades do projeto. O Código 1 é adequado para coleções de dados menores onde a ordem é importante, enquanto o Código 2 é mais robusto e recomendado para cenários onde o desempenho previsível é crucial, especialmente com grandes conjuntos de dados. A Árvore AVL oferece vantagens significativas em termos de desempenho e equilíbrio automático, tornando-a uma escolha sólida para muitos casos de uso.