

BRENDA PINHEIRO RICCI
JOÃO PEDRO VIEIRA SANTOS

ÁRVORE AVL



Trabalho apresentado ao Programa de Graduação em Ciência da Computação, Setor de Ciências Exatas, Universidade Federal do Paraná, disciplina de Algoritmos e Estrutura de Dados III.

Professor:

EDUARDO ALMEIDA

Nota:

Curitiba

Dezembro/2022

Introdução

O presente relatório tem como objetivo descrever os métodos utilizados para a implementação da Árvore AVL, como solicitado na descrição do trabalho.

1. Sobre a Árvore AVL

A Árvore AVL é um tipo de árvore binária de busca. As árvores binárias consistem em um nó que pode ter dois, um ou nenhum nó filho, sabendo que cada um desses nós também pode ser uma árvore binária. Além disso, as chaves dos nós (ou valores deles) seguem um padrão: o nó da esquerda sempre é menor que o nó pai e o nó da direita é sempre maior que o nó da direita.

Entretanto, as AVLs se diferenciam por uma regra específica: elas são sempre balanceadas. Isso significa que a diferença de altura entre a árvore direita e esquerda nunca pode passar de ± 1 .

2. Criação da struct

O primeiro passo para a implementação da AVL é a criação da struct contendo o nodo, o balanceamento e os nodos filhos (esquerdo e direita). A princípio os nodos filhos são apontados para NULL e o balanceamento é 0.

3. Funções principais

- **Inserir Nodo:** A função `inserirAVL` insere um novo nodo, percorrendo a árvore até encontrar o local adequado para o novo nodo de forma recursiva, além disso também verifica o balanceamento e realiza a rotação se necessário.
- **Deleta Nodo:** A função `deletarNodo` percorre a AVL para encontrar o nodo com a chave necessária e deleta esse nodo, além disso também verifica o balanceamento e realiza a rotação caso necessário.
- **Verifica se é AVL:** A função `ehAVL` verifica o balanceamento de toda a árvore.
- **Rotações:** As funções `rotacaoDir` e `rotacaoEsq` efetuam a rotação para a direita e esquerda respectivamente.

4. Funções extras

Entre as funções extras que foram utilizadas para facilitar as funções principais estão:

- **max:** retorna o máximo de dois números

- emOrdem : imprime os valores da árvore em ordem crescente
- preOrdem: imprime a raiz da árvore, e seus filhos da esquerda e direita respectivamente.
- posOrdem: imprime os filhos da esquerda e da direita de uma árvore respectivamente.
- alturaArvore: calcula e retorna a altura da árvore
- imprimeSaída: Imprime a árvore indicando as chaves e a altura do nodo.

5. Testes

Para verificar o funcionamento do algoritmo descrito, foram realizados alguns testes (Teste 1 e Teste 2, respectivamente):

Entrada	Saída
<pre> 1 i 10 2 i 20 3 i 30 4 i 40 5 i 50 6 i 45 7 i 48 </pre>	<pre> 1 10,2 2 20,1 3 30,2 4 40,0 5 45,2 6 48,1 7 50,2 </pre>
<pre> 1 i 10 2 i 20 3 i 30 4 i 40 5 i 50 6 i 45 7 i 48 8 r 40 </pre>	<pre> 1 10,2 2 20,1 3 30,2 4 45,0 5 48,1 6 50,2 </pre>

6. Conclusão

Com base nos processos descritos e nos resultados obtidos nos testes, pode-se dizer que o programa de implementação da Árvore AVL funciona como o esperado