

# **Estrutura de Dados**

## **Relatório Final**

### **Árvore Binária de Busca AVL**

**Gustavo Diel<sup>1</sup>, James Neuburger Gaston<sup>1</sup>, Marlon Henry Schweigert<sup>1</sup>**

<sup>1</sup>Centro de Ciências Tecnológicas – Universidade do Estado de Santa Catarina (UDESC)  
Joinville – SC – Brazil

**Resumo.** *Este relatório descreve o funcionamento do algoritmo de árvore binária de busca AVL (balanceada), exigido pelo professor Doutor Gilmário Barbosa dos Santos como trabalho final da matéria de Estruturas de Dados de Bacharelado de Ciência da Computação.*

## **1. Introdução**

O algoritmo em questão, tem por objetivo, facilitar a busca de dados que são manipulados dinamicamente, bem como inserção e remoção. O seu desenvolvimento foi realizado pelos acadêmicos com o objetivo de estudar o desempenho, viabilidade e praticidade para manipulação e busca de dados em quantidade massiva.

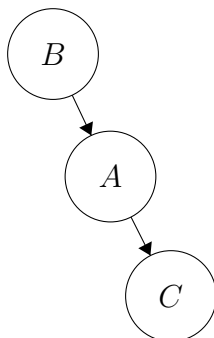
## **2. Algoritmo**

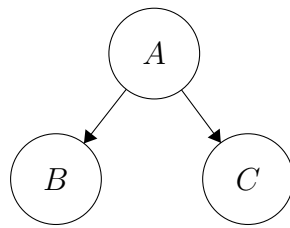
Árvore AVL é uma árvore autobalanceada utilizada para otimizar busca de dados criada por Adelson Velsky e Landis em 1962.

A árvore binária tem por regra básica que seu filho a direita sempre será menor que o pai; consecutivamente, seu filho a esquerda sempre será maior que o pai. Além disso, a árvore é uma árvore autobalanceada e possui uma característica para otimizar a busca: a maior altura pela esquerda não pode se diferenciar bruscamente da árvore a direita. Essa diferença não pode ser maior que 1 em módulo.

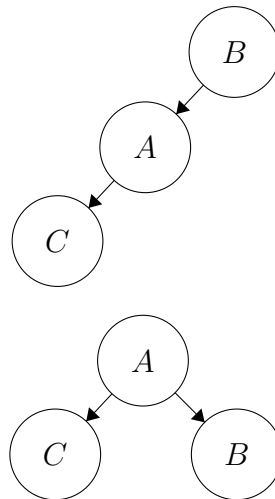
Para manter o balanceamento da árvore, efetuamos as chamadas rotações. Basicamente, são duas rotações simples que existem. A exemplo, a rotação esquerda simples efetua o balanceamento se o filho e neto estão à direita e o seu fator de balanceamento portanto é igual a -2. Da mesma forma, existe a rotação direita que funciona de maneira semelhante, porém espelhado com fator de balanceamento igual a 2.

Rotação Esquerda Simples



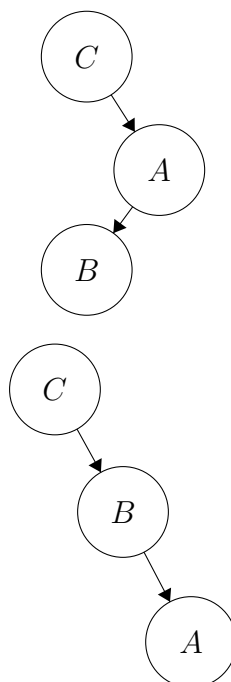


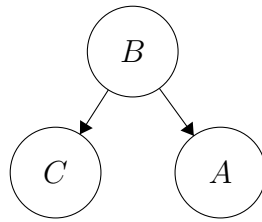
Rotação Direita Simples



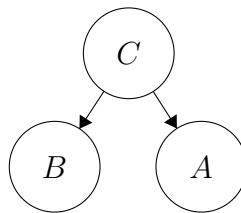
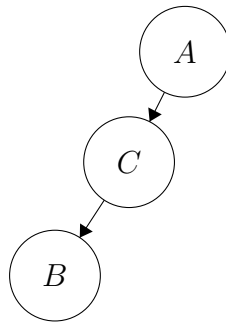
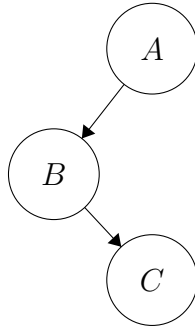
Outras formas de rotação são as rotações duplas. Neste caso a rotação esquerda dupla efetua o balanceamento se o filho está à direita e o neto à esquerda e o seu fator de balanceamento portanto é igual a -2, rotacionado primeiro o neto com o filho, caindo então no caso de rotação simples. Da mesma forma, existe a rotação direita dupla que funciona de maneira semelhante, porém espelhado com fator de balanceamento igual a 2.

Rotação Esquerda Dupla





Rotação Direita Dupla



### 3. Aplicação

A aplicação requerida como trabalho necessita ler um arquivo, mapea-lo (index) e salvar os dados contidos na árvore. Após isso, será possível realizar as seguintes operações:

1. Buscar;
2. Exibir registro completo;
3. Remoção do Registro;
4. Inserção de Registro;
5. Tamanho do Arquivo;
6. Salvar índice em Disco;
7. Recuperar do índice a partir do Disco;

### **3.1. Buscar**

Na aplicação, foram construídos dois tipos de busca, uma pelo índice e pela estrutura de dados salvo na árvore.

Pela alternativa de exibir a linha do arquivo, é efetuado uma busca pela matrícula do funcionário, após isso buscando no arquivo pelo valor do seu índice.

Pela alternativa de exibir o dado, é exibido o dado que está na árvore (Matrícula, salário, nome, telefone, etc...).

### **3.2. Remoção e Inserção do Registro**

A inserção de um dado é feito diretamente na árvore. Após preenchido os dados requeridos no formulário, o método de inserção em árvore AVL insere a estrutura criada pela aplicação na Estrutura de Dados.

A remoção de dados ocorre também pela TDA da árvore. Os dados são removidos da memória.

### **3.3. Salvar índice em Disco**

Ao salvar a árvore em disco, estará alterando o arquivo de entrada de dados em forma "inOrder", otimizando o carregamento de dados sem necessitar as rotações na próxima inicialização.

## **4. Considerações Finais**

O código se mostrou responsivo e eficaz para a resolução do problema proposto. O método "inOrder" para releitura da árvore mostrou-se eficiente ao inicializar a aplicação.

A Árvore AVL mostrou-se veloz em sua consulta e manipulação dos dados, mesmo sendo uma estrutura de dados complexa.