

# **Relatório Técnico**

## **Pablo Duarte da Silva**

### **Resumo do Projeto**

O Relatório técnico apresentado, refere-se a descrição dos conceitos, métodos e soluções utilizados para a resolução da primeira avaliação de Estruturas de Dados II.

#### **1. Introdução**

As estruturas de dados são amplamente utilizadas em softwares complexos, principalmente aqueles que necessitam realizar intensas operações de busca e inserção. Esse relatório técnico apresenta um experimento comparativo entre estruturas de dados baseadas em árvores, unicamente com o propósito de medir a eficiência em realizar operações computacionais em tais estruturas. Esse relatório técnico está organizado da seguinte forma. Na Seção 2 serão apresentados alguns conceitos básicos dos assuntos abordados. A Seção 3 contém a metodologia implementada para a realização dos experimentos. A seção 4 aborda os resultados e os dados comparativos. A seção 5 apresenta as conclusões do relatório.

#### **2. Seções Específicas**

##### **2.1. Estrutura de Árvore**

As estruturas de dados de árvores são estruturas nomeadas assim por apresentarem um aspecto representativo semelhante a árvores da vida real. Diferente das estruturas de listas encadeadas, aqui as informações não são dispostas em forma sequencial.

##### **2.2. Árvore Binária**

Em Ciência da computação, uma árvore binária de busca (ou árvore binária de pesquisa) é uma estrutura de dados de árvore binária baseada em nós, onde todos os nós da subárvore esquerda possuem um valor numérico inferior ao nó raiz e todos os nós da subárvore direita possuem um valor superior ao nó raiz (esta é a forma padrão, podendo as subárvores serem invertidas, dependendo da aplicação).

##### **2.3. Árvore AVL**

É uma árvore binária de busca balanceada, ou seja, uma árvore balanceada (árvore completa) são as árvores que minimizam o número de comparações efetuadas no pior caso para uma busca com chaves de probabilidades de ocorrências idênticas. Contudo, para garantir essa propriedade em aplicações dinâmicas, é preciso reconstruir a árvore para seu estado ideal a cada operação sobre seus nós (inclusão ou exclusão), para ser alcançado um custo de algoritmo com o tempo de pesquisa tendendo a  $O(\log n)$ .

#### **3. Metodologia**

Essa seção apresenta a metodologia e ambiente utilizados para realizar a análise e comparação das estruturas de dados citadas. Para o ambiente de testes, foi utilizado um computador para a execução de algoritmos e medição de dados. As configurações de hardware do dispositivo são apresentadas na Tabela 1.

Dispositivo	Processador	Memória RAM	Sistema Operacional
Notebook Acer	AMD A12-9720P 2.7GHz	8 GB	Ubuntu Linux 18.04 LTS

Tabela 1. Configurações de Hardware

O experimento consistiu em submeter o dispositivo a execução de 4 algoritmos de estruturas de dados em árvores Binárias e árvores AVL. Dois algoritmos eram voltados a inserção e busca de valores inteiros, enquanto que os outros dois algoritmos tinham como objetivo inserção e busca em uma árvore de strings.

### 3.1. Funcionalidades Utilizadas

Algumas das funções implementadas aos algoritmos:

1. **treeInitialize:** Função que inicializa uma árvore;
2. **treeInsert:** Recebe a raiz e um novo elemento, insere o novo elemento em seu devido lugar na árvore;
3. **treeSearch:** Recebe o valor de elemento e verifica se o mesmo existe na árvore;
4. **treeHeigth:** Recebe a raiz da árvore e retorna a sua altura;
5. **treeRemove:** Recebe a raiz da árvore
6. **treeRotationLL / treeRotaionRR / treeRotationLR / treeRotationRL:** Funções utilizadas pelos algoritmos de árvore AVL para rotacionar a árvore;
7. **treeHeightOf / treeBalancingFactor / treeMin:** Funções auxiliares utilizadas pelo algoritmo de árvore AVL para balancear a árvore;

## 4. Resultados da Execução do Programa

Essa seção apresenta os resultados obtidos nos experimentos que tiveram como objetivo comparar a eficácia na utilização nas estruturas de dados de árvores apresentadas.

### 4.1 Programa de Busca de Inteiros

Neste experimento, foi utilizado como métrica: a variação de tempo(em nanosegundos) de busca de elementos numa árvore preenchida aleatoriamente.

	Altura 15		Altura 20	
N Buscados	Árvore Binária	Árvore AVL	Árvore Binária	Árvore AVL
# 2252	2,73E+08	2,65E+08	2,68E+06	2,39E+08
# 46445	4,70E+06	2,90E+08	3,10E+06	3,70E+06
# 57861	1,90E+06	4,10E+06	3,00E+06	3,30E+06
# 97962	1,80E+06	2,10E+06	2,30E+06	3,00E+06
# 64556	1,80E+06	2,00E+06	2,60E+06	2,80E+06
# 6229	1,80E+06	1,50E+06	2,10E+06	2,30E+06
# 41059	1,60E+06	6,70E+06	2,20E+06	5,60E+06
# 11870	1,50E+06	1,60E+06	2,20E+06	2,60E+06
# 45857	1,50E+06	1,60E+06	2,00E+06	2,20E+06
# 101761	1,60E+06	1,70E+06	2,20E+06	2,80E+06

Tabela 2. Tempos de Busca nas Árvores

#### 4.2 Programa de Referência Cruzada

Neste experimento, foi realizado para analisar a quantidade de passos necessários para encontrar elementos dentro de uma árvore binária de busca em uma árvore AVL. A medição foi realizada em um programa de referência cruzada, onde o conteúdo da árvore é formado por strings, inseridas a partir da leitura de um arquivo de texto.

	Árvore Binária	Árvore AVL
Palavras Buscadas	Passos Percorridos	Passos Percorridos
querer	7	6
doer	10	7
descontente	8	5
servir	10	7
preso	7	5
ver	6	3
pode	8	6
mesmo	10	7
amor	1	4
amizade	4	7

Tabela 3. Tempos de Busca nas Árvores

#### 5. Conclusão

Esse relatório apresentou uma análise relativa às estruturas de dados baseadas em árvores Binária de Busca e AVL, com o objetivo de comparar a eficácia de ambas as estruturas nas situações apresentadas. Na primeira etapa foi observado o tempo de busca em árvores de valores inteiros. O segundo experimento foi realizado visando medir o número de passos para buscar elementos em árvores de string. Em ambos os casos os resultados relacionados a árvore AVL se mostraram mais estáveis, apresentando um range de variação de tempo e número de passos menor que na árvore Binária de Busca, levando em conta que a árvore AVL há muito mais elementos até alcançar a altura 15 ou 20, por balancear a árvore a cada novo elemento inserido. Com base nos dados apresentados, é possível afirmar que a árvore AVL é uma opção viável para tratar de grandes volumes de dados, dada sua estabilidade. A árvore binária, por sua vez, apesar da sua instabilidade é uma opção recomendável para volumes de dados menores e que necessitam de um tempo de busca menor.

#### 6. Apêndice

Todo código-fonte apresentado para as soluções das questões propostas seguem em anexo com a documentação.

## 7. Referências

ÁRVORE AVL. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2020. Disponível em: <[https://pt.wikipedia.org/w/index.php?title=%C3%81rvore\\_AVL&oldid=59445504](https://pt.wikipedia.org/w/index.php?title=%C3%81rvore_AVL&oldid=59445504)>. Acesso em: 25 set. 2020.

ÁRVORE BINÁRIA DE BUSCA. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2021. Disponível em: <[https://pt.wikipedia.org/w/index.php?title=%C3%81rvore\\_bin%C3%A1ria\\_de\\_busca&oldid=60629462](https://pt.wikipedia.org/w/index.php?title=%C3%81rvore_bin%C3%A1ria_de_busca&oldid=60629462)>. Acesso em: 11 mar. 2021.