



CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS

# Relatório II LAED II

AUTOR:

Pedro Henrique Maia Duarte

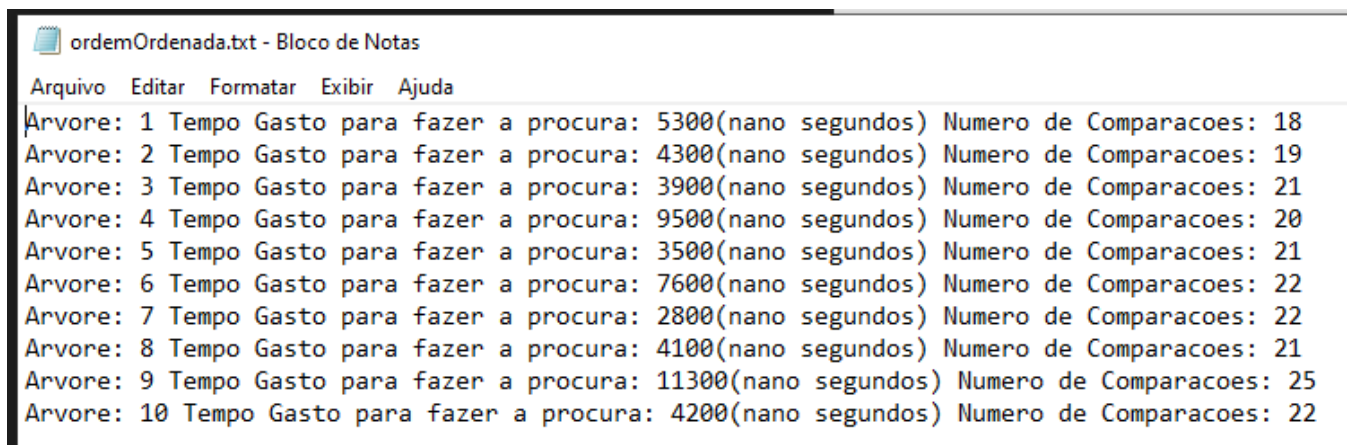
Professor: THIAGO DE SOUZA RODRIGUES

## Prática 2 – Implementação do TAD

### Symetric Binary B-Tree (SBB)

A Árvore SBB é uma árvore binária com apontadores verticais e horizontais, tais que todos os caminhos da raiz até cada nó externo possuem o mesmo número de apontadores verticais e não podem existir dois apontadores horizontais sucessivos. É uma estrutura que minimiza o tempo médio de pesquisa, assumindo distribuição uniforme das chaves, entretanto, manter a árvore completamente balanceada após cada inserção é muito caro.

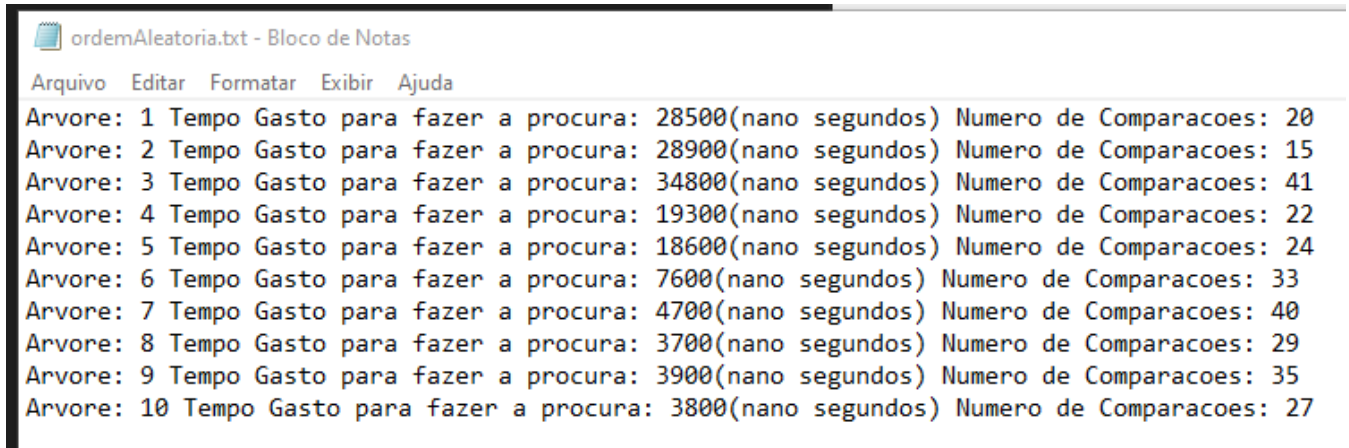
Sendo assim, implementaremos em linguagem Java, um algoritmo de inserção de itens ordenadamente e aleatoriamente, bem como um método de busca de um determinado item em ambas as árvores, com o intuito de analisar a quantidade de comparações realizadas e o tempo gasto em cada operação de busca.



| Arquivo    | Editar   | Formatar                  | Exibir | Ajuda |
|------------|--|---------------------------|--------|-------|
| Arvore: 1  | Tempo Gasto para fazer a procura: 5300(nano segundos)  | Numero de Comparacoes: 18 |        |       |
| Arvore: 2  | Tempo Gasto para fazer a procura: 4300(nano segundos)  | Numero de Comparacoes: 19 |        |       |
| Arvore: 3  | Tempo Gasto para fazer a procura: 3900(nano segundos)  | Numero de Comparacoes: 21 |        |       |
| Arvore: 4  | Tempo Gasto para fazer a procura: 9500(nano segundos)  | Numero de Comparacoes: 20 |        |       |
| Arvore: 5  | Tempo Gasto para fazer a procura: 3500(nano segundos)  | Numero de Comparacoes: 21 |        |       |
| Arvore: 6  | Tempo Gasto para fazer a procura: 7600(nano segundos)  | Numero de Comparacoes: 22 |        |       |
| Arvore: 7  | Tempo Gasto para fazer a procura: 2800(nano segundos)  | Numero de Comparacoes: 22 |        |       |
| Arvore: 8  | Tempo Gasto para fazer a procura: 4100(nano segundos)  | Numero de Comparacoes: 21 |        |       |
| Arvore: 9  | Tempo Gasto para fazer a procura: 11300(nano segundos) | Numero de Comparacoes: 25 |        |       |
| Arvore: 10 | Tempo Gasto para fazer a procura: 4200(nano segundos)  | Numero de Comparacoes: 22 |        |       |

*Dados obtidos por inserção em ordem Ordenada - Imagem 01*

## CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS



|            |  |                           |
|------------|--|---------------------------|
| Arvore: 1  | Tempo Gasto para fazer a procura: 28500(nano segundos) | Numero de Comparacoes: 20 |
| Arvore: 2  | Tempo Gasto para fazer a procura: 28900(nano segundos) | Numero de Comparacoes: 15 |
| Arvore: 3  | Tempo Gasto para fazer a procura: 34800(nano segundos) | Numero de Comparacoes: 41 |
| Arvore: 4  | Tempo Gasto para fazer a procura: 19300(nano segundos) | Numero de Comparacoes: 22 |
| Arvore: 5  | Tempo Gasto para fazer a procura: 18600(nano segundos) | Numero de Comparacoes: 24 |
| Arvore: 6  | Tempo Gasto para fazer a procura: 7600(nano segundos)  | Numero de Comparacoes: 33 |
| Arvore: 7  | Tempo Gasto para fazer a procura: 4700(nano segundos)  | Numero de Comparacoes: 40 |
| Arvore: 8  | Tempo Gasto para fazer a procura: 3700(nano segundos)  | Numero de Comparacoes: 29 |
| Arvore: 9  | Tempo Gasto para fazer a procura: 3900(nano segundos)  | Numero de Comparacoes: 35 |
| Arvore: 10 | Tempo Gasto para fazer a procura: 3800(nano segundos)  | Numero de Comparacoes: 27 |

*Dados obtidos por inserção em ordem Aleatória – imagem 02*

Quando inserimos na árvore elementos ordenadamente, o elemento que contém o maior valor será o último componente inserido na árvore 10, onde terá como conteúdo a chave 100000. Ademais, quando inserimos na árvore elementos aleatoriamente, o elemento que pode conter a chave 100000. Sendo assim, para testar o tempo de busca, iremos procurar em todas as árvores criadas, um elemento cujo valor é 200000, para garantir que uma extremidade da árvore será percorrida totalmente.

Sendo assim, pretendemos analisar quantas comparações de elementos são realizadas em cada árvore, e esperamos que nas árvores de inserção ordenada, o número de comparações seja próximo, pois se trata de uma árvore balanceada. Enquanto, isso, esperamos que o número de comparações em árvores com inserção aleatória seja parecido com a inserção ordenada, pois a árvore SBB, deve se manter balanceada.

Logo, temos o seguinte gráfico:

### Número de comparações entre as árvores

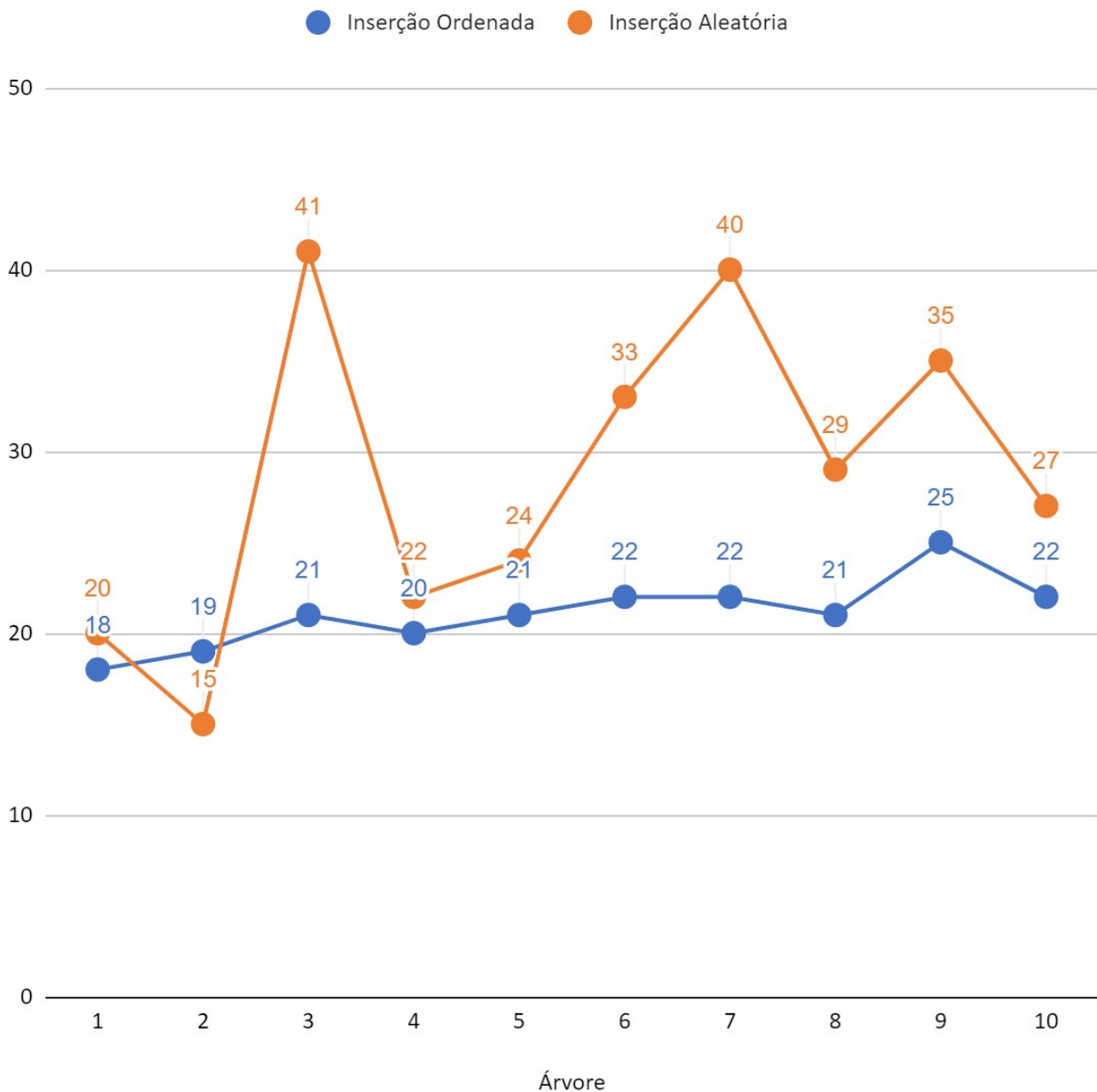


Gráfico 01

O gráfico apresentou os resultados esperados, garantindo que todos os elementos foram inseridos nas árvores, e conseqüentemente, em ambos os casos, foram comparados com o valor de 100000, e não foi encontrado a chave procurada. Sendo assim, temos um gráfico com valores próximos, e apesar do número alto de elementos inseridos, a quantidade de comparação é próxima ao 0. Sendo assim,

## CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS

percebemos que a inserção utilizando o método de logaritmo, utilizado para a construção destas árvores, é superior em relação a inserção linear. Além disso, temos a relação de tempo gasto durante a procura:

### Tempo de procura(ns) entre as árvores

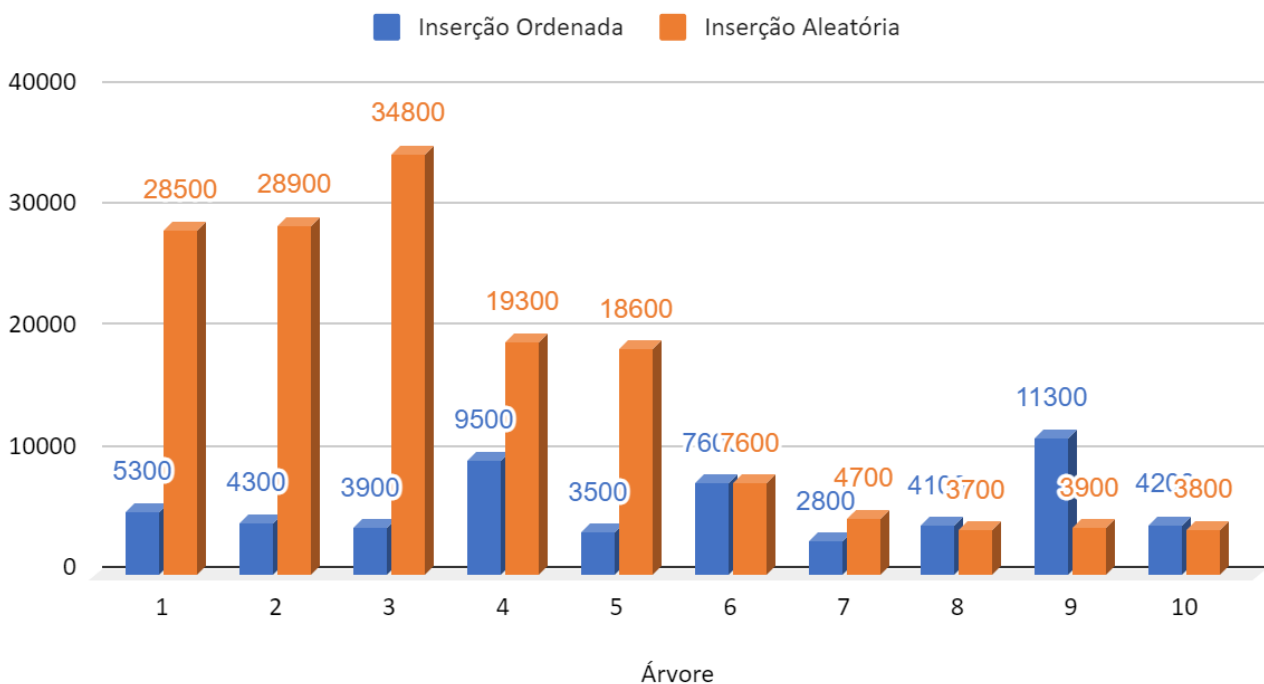


Gráfico 02

Analisando o gráfico, percebemos que a procura em árvores com menos elementos, levou um tempo maior que a busca em árvores com mais elementos, em inserção aleatória. Logo, percebemos que o gráfico contém dados discrepantes.

Logo, concluímos que a grandeza tempo de relógio não é capaz para inferir em uma conclusão de complexidade de busca, por apresentar resultados imprecisos e discrepantes, atuando como apenas uma informação e sem produzir uma relevância direta. Sendo assim, o número de comparações é mais adequado para analisar a complexidade de busca, onde podemos comparar o número de comparações executadas para cada árvore para obter o mesmo resultado, como nesse caso, é procurar um elemento que nunca estará presente.

Além disso, podemos analisar com os resultados obtidos perante a última prática, sobre Árvores Binárias de Pesquisa:

- Inserção Ordenada:

Na Árvore Binária de Pesquisa obtemos um gráfico linear sobre o número de comparações realizadas entre as árvores para um elemento não existente na árvore, enquanto na Árvore SBB, tivemos

## CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS

resultados próximos para todas as árvores. Sendo assim, concluímos que para a inserção ordenada, a Árvore SBB por ser balanceada apresenta resultados superiores em relação a Árvore Binária de Pesquisa, pois precisou fazer menos comparações para definir uma resposta.

Em relação ao tempo de máquina, em ambos os casos tivemos resultados discrepantes entre si, portanto, não é possível afirmar nada sobre o tempo de máquina aplicado em ambas as árvores.

### - Inserção Aleatória:

Em termos sobre o número de comparações realizadas, em ambas as árvores, os gráficos apresentam características semelhantes.

Em relação ao tempo de máquina, em ambos os casos tivemos resultados discrepantes entre si, portanto, não é possível afirmar nada sobre o tempo de máquina aplicado em ambas as árvores.

Sendo assim, em termos de análise geral do comportamento de todos os gráficos, concluímos que a Árvore SBB possui resultados superiores em relação ao número de comparações realizadas para definir que um elemento não está presente na árvore é superior do que a Árvore Binária de Pesquisa. Isso comprova que o gasto em comparações  $O(\log(n))$  é superior ao  $O(n)$ .