

CEFET-MG - Campus II  
Departamento de Engenharia de Computação  
Laboratório de Algoritmos e Estruturas de Dados II

## Prática III

*Implementação em JAVA do TAD ArvoreB*

Alunos: Antônio Augusto Diniz Sousa  
Professor orientador: Thiago de Souza Rodrigues

09 de Setembro  
2018

CEFET-MG - Campus II  
Departamento de Engenharia de Computação  
Laboratório de Algoritmos e Estruturas de Dados II

## Prática III

### *Implementação em JAVA do TAD ArvoreB*

Relatório da prática III apresentado à Disciplina de Laboratório de Algoritmos e Estruturas de Dados II do Curso de Engenharia de Computação do Centro Federal de Educação Tecnológica de Minas Gerais, como requisito parcial para conclusão da disciplina.

Alunos: Antônio Augusto Diniz Sousa

Professor orientador: Thiago de Souza Rodrigues

09 de Setembro  
2018

# Sumário

<b>1</b>	<b>Apresentação</b>	<b>1</b>
<b>2</b>	<b>Descrição de atividades</b>	<b>2</b>
2.1	Análise . . . . .	3
<b>3</b>	<b>Conclusão</b>	<b>4</b>

# 1 Apresentação

A tarefa dessa prática foi testar o comportamento da Árvore B com diversas ordens e em diversos casos. Utilizou-se a implementação do Autor Ziviani, disponível nesse link ou acessado diretamente em <http://www2.dcc.ufmg.br/livros/algoritmos-java/implementacoes-05.php>.

Utilizando a linguagem *Java*, criou-se 40 árvores, sendo 10 delas utilizando a modelagem SBB, e as outras 30, tipo B, variando a ordem entre 2, 4 e 6. Dentre as dez de cada modelo, a primeira possuía 10000 elementos, a segunda 20000, e assim por diante, até a 10<sup>a</sup>, com 100000 elementos.

Analisou-se diante dessas árvores o número de comparações necessárias para buscar um elemento não existente nas árvores e, nos testes feitos nas árvores B, também foi analisado quantas páginas foram acessadas durante essa busca. Os resultados seguem adiante nesse relatório e os códigos fontes podem ser acessados clicando aqui ou acessando em <https://gitlab.com/antonioaads/LAEDII>.

## 2 Descrição de atividades

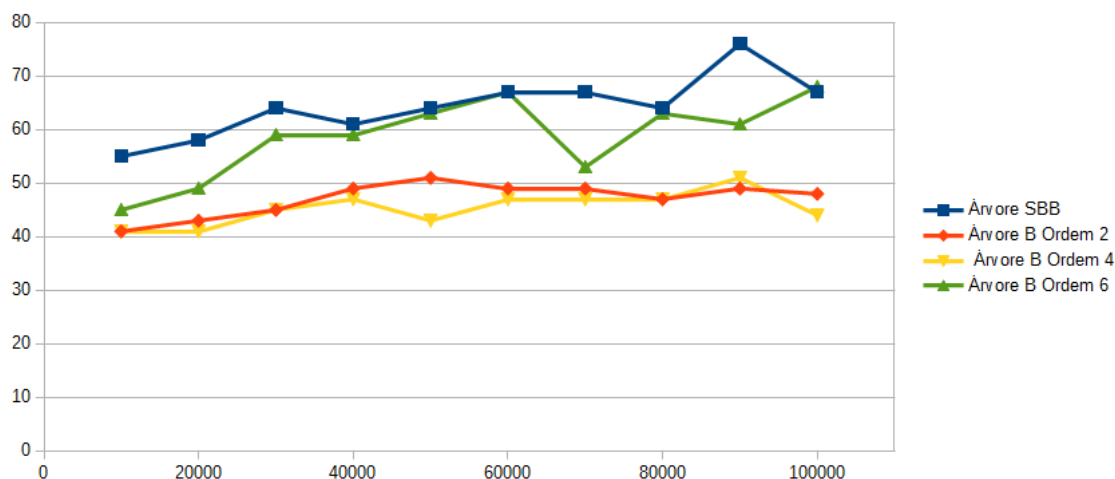
Após a execução do código, que não iremos tratar especificamente nesse relatório, pois o mesmo se encontra no repositório citado na introdução devidamente comentado e organizado, tirou-se os resultados da Tabela 1.

Tabela 1: Resultados

Itens	SBB	B Ordem 2		B Ordem 4		B Ordem 6	
	Comp.	Comp.	Pág.	Comp.	Pág.	Comp.	Pág.
10000	55	41	9	41	9	45	6
20000	58	43	10	41	7	49	6
30000	64	45	10	45	7	59	6
40000	61	49	11	47	8	59	7
50000	64	51	11	43	8	63	7
60000	67	49	11	47	8	67	7
70000	67	49	11	47	8	53	7
80000	64	47	11	47	8	63	7
90000	76	49	11	51	8	61	7
100000	67	48	10	44	7	68	6

Utilizou-se a ferramenta *LibreOffice Calc* para plotar o gráfico ilustrados na Figura 1.

Figura 1: Representação do gráfico  $n \times$  Número de comparações



## 2.1 Análise

A Árvore Binária SBB possui uma complexidade de tempo para busca  $O(\log n)$  tanto para o **caso médio** quanto para o **pior caso**, como já mostrado na prática anterior, disponibilizada no mesmo repositório onde encontra-se esta.

A árvore B também possui complexidade  $O(\log n)$  para busca, o que nos leva a pensar o que devemos considerar para escolher uma ou outra implementação.

Observando o gráfico ilustrado na Figura 1, podemos analisar que todas as ondas possuem um formato semelhante a uma curva logarítmica, porém com um pequeno *offset* entre uma e outra.

Percebemos que todas as árvores B, foram, na maioria dos casos, mais eficiente que a Árvore SBB, o que já justifica uma implementação dessa TAD. Agora, entre as Árvore B implementadas, a de ordem 2, 4 e 6, também demonstraram resultados distintos, que varia muito de aplicação para aplicação. Para a aplicação testada, vetor devidamente ordenado, a árvore de ordem 4 demonstrou-se a mais eficiente.

### 3 Conclusão

A Árvore B possui uma complexidade interessante para todas as operações, porém, possui um código um pouco mais complexo e extenso.

A escolha depende, como todas as árvores já tratada em outras práticas, da aplicação, e deve ser escolhida minunciosamente, analisando custo para produção do algoritmo, custo de implementação, dentre outros diversos parâmetros.