

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

Prática I

*Implementação em JAVA do TAD Árvore
Binária de Pesquisa*

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

26 de Agosto
2018

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

Prática I

*Implementação em JAVA do TAD Árvore
Binária de Pesquisa*

Relatório da prática I 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

26 de Agosto
2018

Conteúdo

1	Apresentação	1
2	Descrição de atividades	2
2.1	Análise	4
3	Conclusão	5

1 Apresentação

A tarefa dessa prática foi testar o comportamento da Árvore Binária de Busca 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 18 arvores, sendo 9 delas utilizando uma inserção de dados ordenados, e as outras 9, injeção de dados aleatórios. Dentre as nove de cada modelo, a primeira possuía 1000 elementos, a segunda 2000, e assim por diante, até a 9ª, com 9000 elementos.

Analisou-se diante dessas árvores o tempo gasto para pesquisar um item ausente na estrutura e as comparações necessárias para tirar tal conclusão. Os resultados seguem adiante nesse relatório e todos os códigos fontes podem ser acessados pelo 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

Número de Itens	Ordenada		Aleatória	
	Comparações	Tempo (ns)	Comparações	Tempo (ns)
1000	3001	103158	25	2140
2000	6001	117283	25	856
3000	9001	131408	25	1284
4000	12001	144678	25	1712
5000	15001	178921	28	1284
6000	18001	295776	28	1284
7000	21001	320602	28	856
8000	24001	371538	28	2141
9000	27001	422903	28	16694

É de conhecimento que a a medida de tempo pode variar por n motivos, sendo alguns,

- Arquitetura da máquina;
- Configuração da máquina;
- Atividades rodando de maneira paralela;
- Modos de economia ou gerenciamento de energia

dentre diversos outros. A fim de especificar melhor, e tentar minimizar erros e diferenças do tempo mensurado devido aos motivos citados, utilizamos o mesmo computador para todos os testes, com especificação da Tabela 2.

Tabela 2: Hardware Utilizado

Modelo	Dell Vostro 3450
Memória RAM	8GB
Sistema Operacional	Ubuntu 18.04 LTS
Processador	Intel Core i5 6 ^a Geração

Utilizou-se a ferramenta o *LibreOffice Calc* para plotar os gráficos ilustrados na Figure 2 e Figure 1.

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

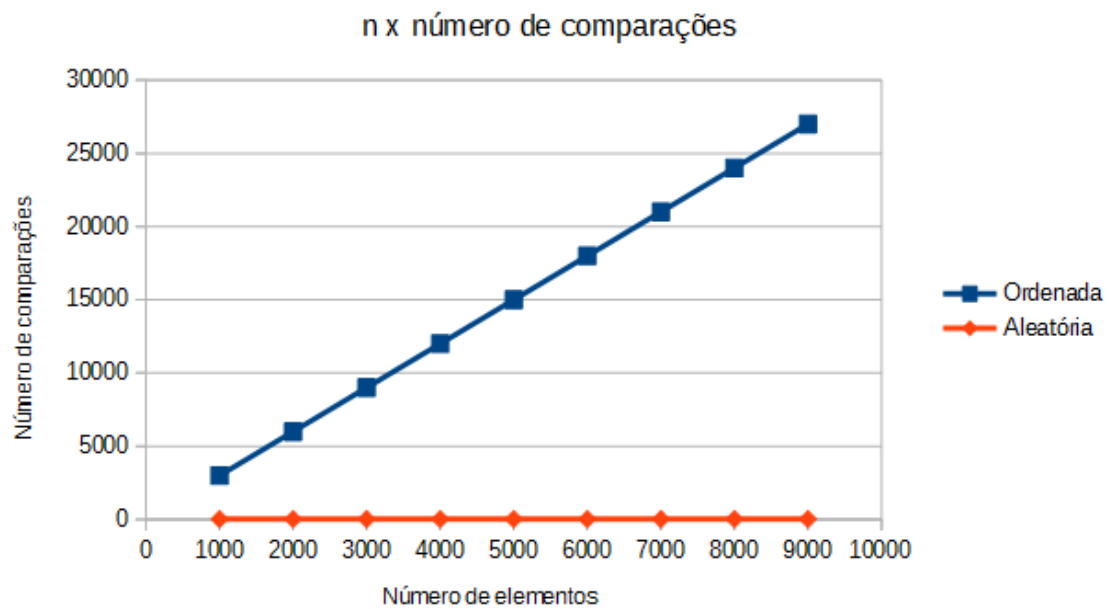
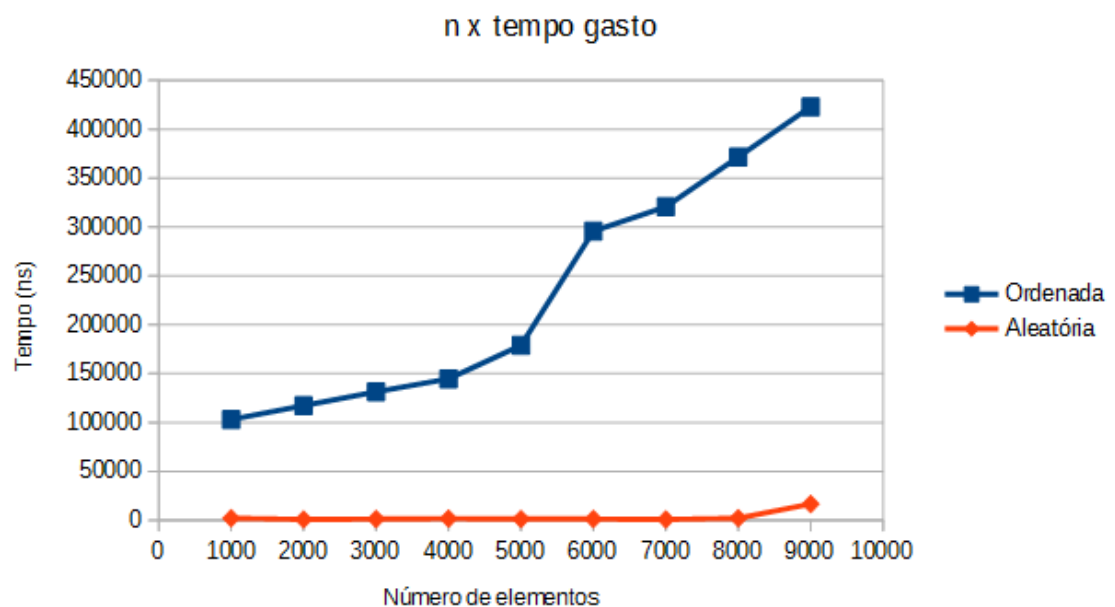


Figura 2: Representação do gráfico $n \times$ Tempo Gasto



2.1 Análise

A Árvore Binária de pesquisa possui uma complexidade de tempo para busca $O(\log n)$ para o **caso médio** e $O(n)$ para o **pior caso**.

Quando inserimos itens na TAD em questão de maneira ordenada, tornamos a estrutura desbalanceada, de maneira que um ramo da árvore fique muito profundo enquanto o outro, fica com poucos ou até mesmo nenhum elemento. Isso faz com que a busca seja linear, semelhante a busca em um vetor, o que torna claro nos gráficos demonstrados da árvore ordenada.

Porém, quando inserimos de maneira aleatória, a Árvore Binária de Busca tende a ficar mais balanceada, minimizando a profundidade da TAD, logo, facilitando encontrar os termos, pois divide-se a quantidade total de itens por dois a cada passo. O resultado disso é uma variação minimalista no número de comparações e do tempo gasto com a variação grotesca da quantidade de termos analisados.

O gráfico de tempo não ficou perfeitamente linear devido as condições citadas no início da sessão, porém foi o suficiente para perceber claramente a alteração da eficiência da TAD com a variação dos termos inseridos.

3 Conclusão

Pode-se concluir diante dessa prática que é necessário analisar minuciosamente a aplicação antes de implementar uma Árvore Binária de Busca, porém, em casos gerais, que os dados são inseridos aleatoriamente, ela demonstra uma alta eficiência mesmo com uma implementação simples.