



## Relatório AEDS II - CEFET-MG

**Alunos:** Bernard Menezes.  
**Prática 2**

25 de agosto de 2017

## Sumário

---

1	Introdução	1
2	Códigos	1
3	Resultados	1
3.1	Comparações de registros . . . . .	2
3.2	Tempo . . . . .	3
4	Conclusão	3

## 1 INTRODUÇÃO

---

Este relatório refere-se à Prática 2 da matéria de Algoritmos e Estrutura de Dados II. Toda a codificação a qual este relatório faz referência foi feita na linguagem de programação JAVA e a IDE Eclipse, utilizando como base as implementações disponíveis nesse [site](#).

Nesta prática foi pedido a implementação de um **árvore SBB** com uma codificação específica para a classe Item, e também foram solicitados os métodos de inicialização, inserção, pesquisa, `pre_order` e métodos auxiliares de movimentações necessárias. Todas as simulações presentes neste documento referem-se a experimentos feitos em um computador com a seguinte arquitetura: processador Intel Atom N470 (512K Cache, 1.83 GHz) 2GB de RAM e 320GB de memória secundária.

## 2 CÓDIGOS

---

Os códigos com os respectivos comentários encontram-se no mesmo repositório que este relatório, e como já dito anteriormente, utilizam-se de implementações disponibilizadas pelo Professor Emérito do Departamento de Ciência da UFMG, Nivio Ziviani, em [site](#). A saída dos programas utilizados para gerar os gráficos presentes neste relatório podem ser encontrados no arquivo: "arvore\_sbb/saida".

## 3 RESULTADOS

---

Como pedido, os elementos foram inseridos de duas formas diferentes na árvore, de maneira ordenada e de maneira aleatória. Após as inserções na árvore o método de pesquisa foi chamado para um elemento que não estava presente nela, e durante essa procura foram realizado dois testes medindo: o número de comparação de registros e de tempo (milissegundos). Os teste foram realizados com o número de elementos a serem inseridos variando de 1000 até 9000 com passo de 1000 elementos. Foram gerados gráficos comparativos para mensurar e diferenciar o desempenho nos dois tipos de inserção.

### 3.1 COMPARAÇÕES DE REGISTROS

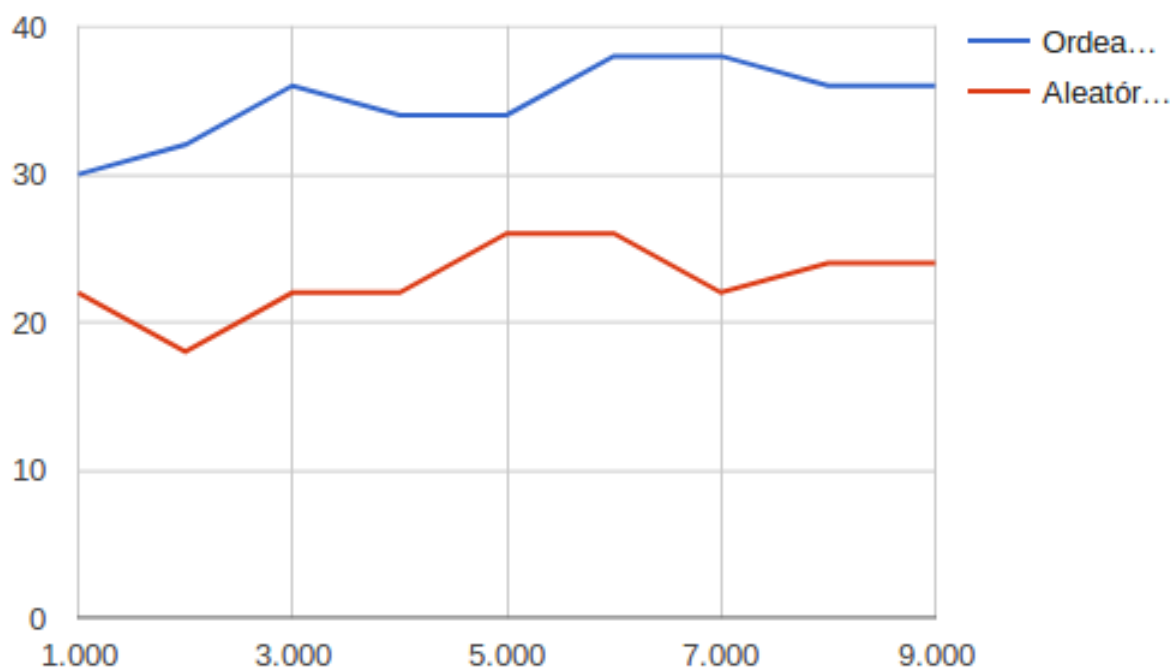


IMAGEM 1: GRÁFICO DE COMPARAÇÕES X TAMANHO

Como a imagem 1 mostra, quando a inserção em uma árvore SBB ocorre de maneira ordenada, o número de comparações entre registros aumenta, porém não muito ao considerarmos a proporção do tamanho da árvore. A pequena diferença entre o número de comparações ocorre devido a capacidade da árvore SBB se modificar de acordo com a inserção de elementos. Existe algumas propriedades a serem obedecidas para que a árvore seja considerada SBB, e caso alguma inserção ou remoção venha a interferir nessas condições necessárias é preciso realizar uma transformação local para manter as propriedades da árvore. Essa propriedades conferem um caráter de árvore "quase balanceada" à árvore SBB, o que indica que em sua totalidade a árvore cresce lateralmente também quando se insere elementos ordenados, ao contrário que se é visto na árvore binária.

## 3.2 TEMPO

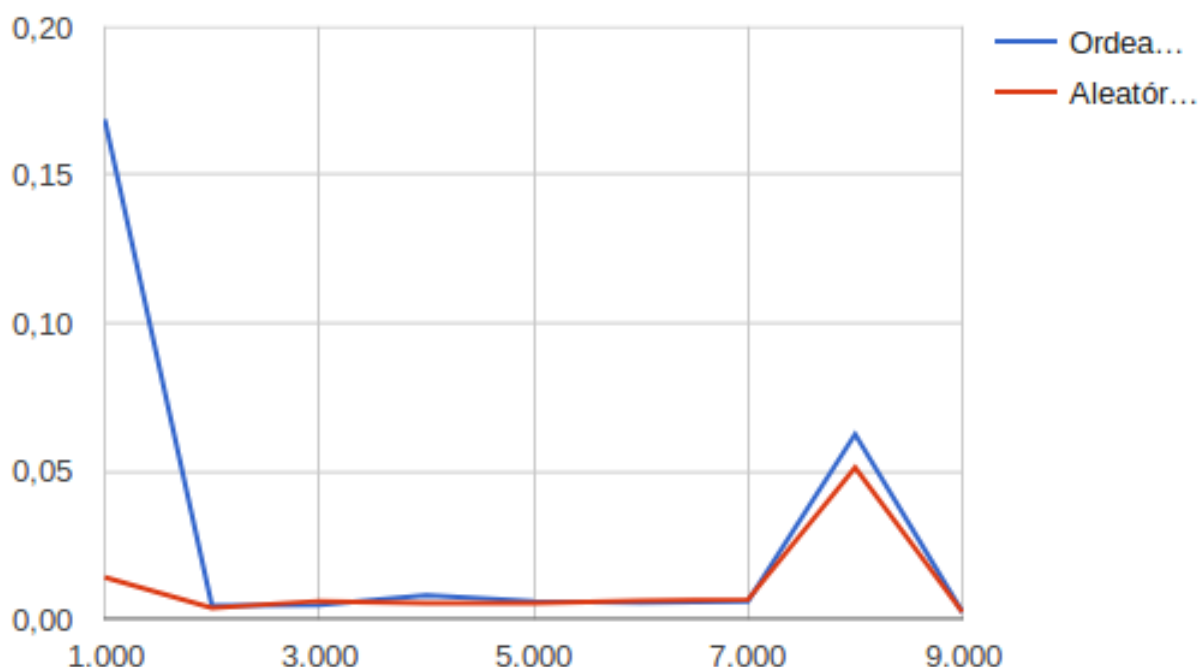


IMAGEM 2: GRÁFICO DE COMPARAÇÕES X TAMANHO

O gráfico 2 é consequência do que foi dito na seção 3.1, como o método de inserção ordenada de elementos gera uma diferença muito pequena no número de comparações de registros, a diferença entre o tempo de execução da árvore que tem elementos inseridos de maneira ordenada e entre o tempo de execução de uma árvore que tem elementos inseridos de maneira aleatória diminui. Por isso a semelhança entre os tempos medidos.

## 4 CONCLUSÃO

Através dessa prática é possível concluir que árvores SBB são eficientes quanto a pesquisa de elementos, pois ela sempre cresce lateralmente devido às suas condições necessárias para ser uma árvore SBB. Entretanto, o custo de inserção é maior, pois pode implicar em diversas transformações locais que demandam mais tempo para inserir um elemento do que uma árvore binária por exemplo.