

# Trabalho de Estrutura de Dados II

Bruno Rafael dos Santos e Luís Bertelli

24 de Julho

## 1 Introdução

O presente trabalho contém um diretório no qual se encontram as pastas: *Graphics*, *AVLTree*, *RedBlackTree* e *BTree* além de um código em python ***PlotResults.py*** e um arquivo de comandos em lote ***RunAll.bat***.

Cada pasta mencionada (com excessão da pasta *Graphics*) possui a implementação da estrutura de dados respectiva ao nome da pasta, e tais implementações quando executadas geram 2 arquivos de texto: *PerformanceAverageCase.txt* e *PerformanceWorstCase.txt* quais correspondem aos testes de caso médio e de pior caso respectivamente, de modo que tais arquivos armazenam em cada linha uma sequência de números correspondentes aos resultados de um teste. Note que cada linha possui 100 números tal que o  $e$ -ésimo número corresponde ao número de etapas para inserção do  $e$ -ésimo elemento à árvore no caso de teste da respectiva linha. Adicionalmente vale destacar que para os testes de caso médio cada elemento inserido na árvore é aleatório enquanto para o pior caso os elementos são inseridos em ordem crescente (os elementos são números de 1 à 100 nesse último caso) e além disso cada implementação possui um arquivo header para encapsulamento das funções e estruturas.

O código ***PlotResults.py*** é utilizado para acessar os arquivos de texto dos testes de todas as estruturas de dados implementadas e gerar gráficos na pasta *Graphics*. Já o arquivo ***RunAll.bat*** serve para executar todos os testes das estruturas de dados implementadas e executar o código ***PlotResults.py*** em seguida.

## 2 Execução dos Testes

Para executar os testes basta que se execute o arquivo ***RunAll.bat***, assim os testes de caso médio irão gerar novos resultados (visto que são feitos com números aleatórios) e novos gráficos serão gerados. A seguir está o link para acesso de todos os arquivos até aqui mencionados: <https://github.com/bruniculos08/EDA2001/tree/main/Trabalho>

## 3 Resultados

Quanto a geração dos resultados deve-se destacar que para obtenção dos resultados para o gráfico de caso médio foram feitos 10 testes, assim cada ponto  $(x, y)$  representa a média dos 10 casos tal que se  $a_{ij}$  é o  $j$ -ésimo elemento da linha  $i$  então:

$$y = \sum_{i=1}^{10} \frac{a_{ij}}{10}$$

e  $x = j$ .

### 3.1 Árvore Rubro-Negra

Para a árvore rubro-negra tivemos os seguintes resultados:

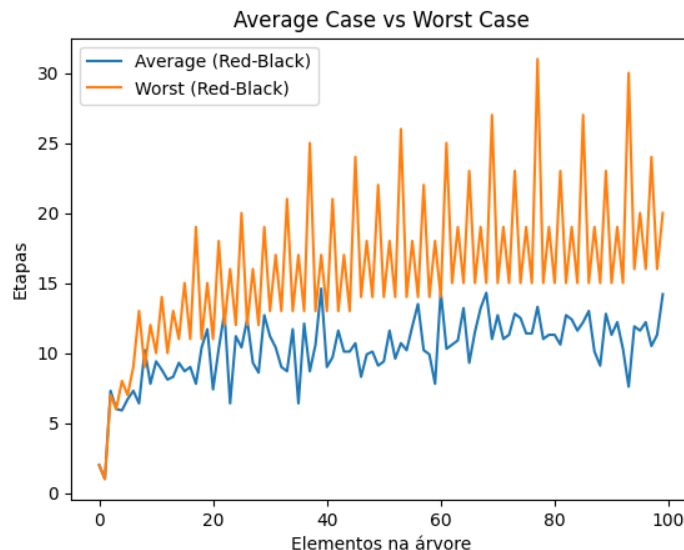


Figura 1: caso médio (em azul) e pior caso (em laranja) de inserção na árvore rubro-negra

### 3.2 Árvore AVL

Para árvore AVL, é importante ressaltar que não foi incluído um contador de etapas para a operação que mede a altura de uma sub-árvore visto que idealmente tal sub-árvore pode ter a altura como um atributo de seu nó raiz, de modo que tal altura é atualizada a cada inserção ao se passar pelo nó, ou seja, idealmente a estrutura do nó deveria ter um atributo de inteiro para a altura da sub-árvore a partir de cada nó. Tendo isso posto, tivemos os seguintes resultados:

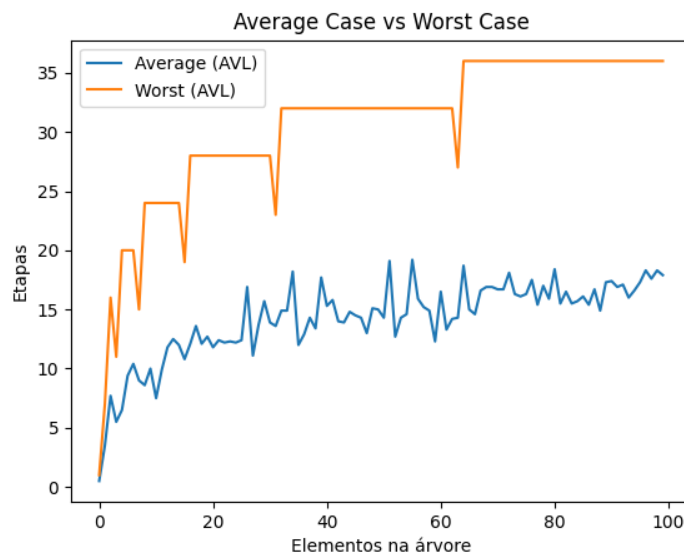


Figura 2: caso médio (em azul) e pior caso (em laranja) de inserção na árvore AVL

### 3.3 Árvore B

Para árvore B tivemos os seguintes resultados:

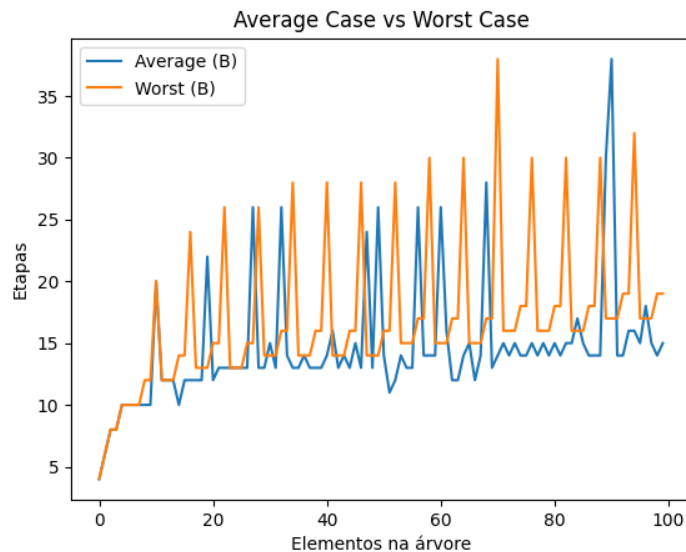


Figura 3: caso médio (em azul) e pior caso (em laranja) de inserção na árvore B

### 3.4 Comparação

Juntando todos os resultados em um único gráfico para maior facilidade na comparação das diferentes estruturas de dados tivemos os seguintes resultados para os casos médios:

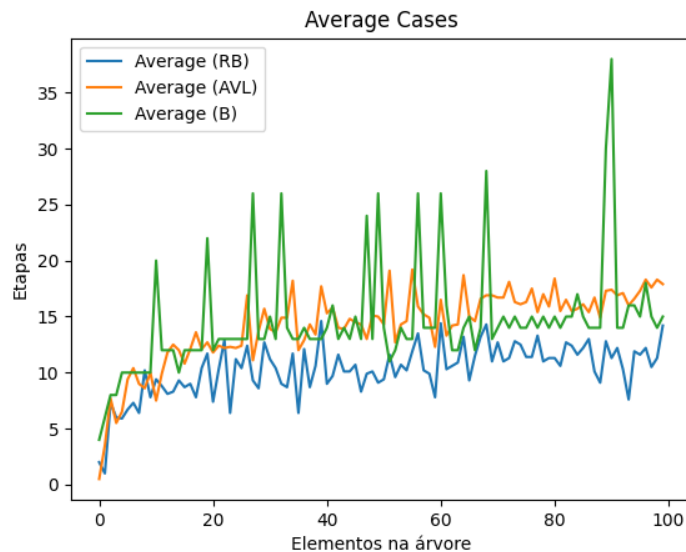


Figura 4: casos médios de inserção em árvores

Enquanto para os piores casos tivemos:

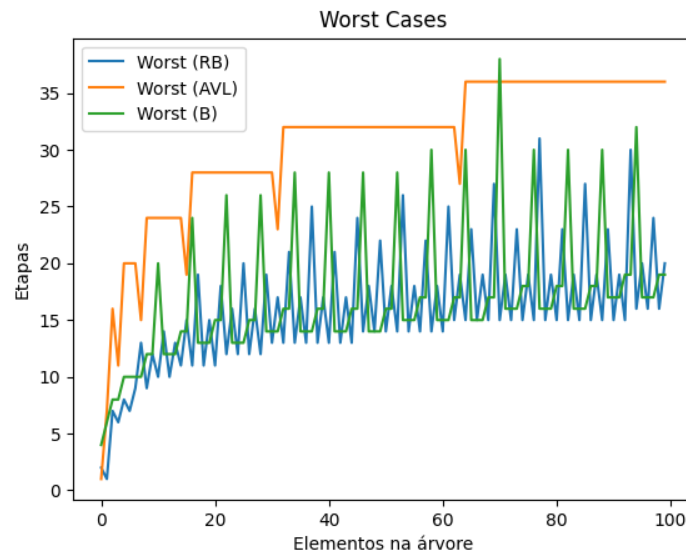


Figura 5: piores casos de inserção em árvores

## 4 Conclusão

Como esperado todos os resultados apresentam gráficos que seguem um padrão logarítmico visto que a complexidade de inserção de todas estruturas de dados testadas é de  $\Theta(\log_2(n))$ . Tal padrão pode ser notado no seguinte gráfico:

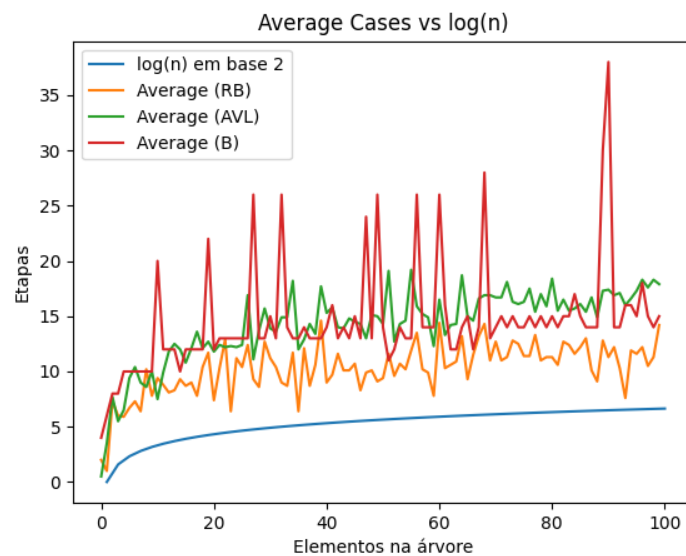


Figura 6: casos de inserção em árvores e função  $\log_2(n)$