

Relatório: desenvolvimento de árvores balanceadas

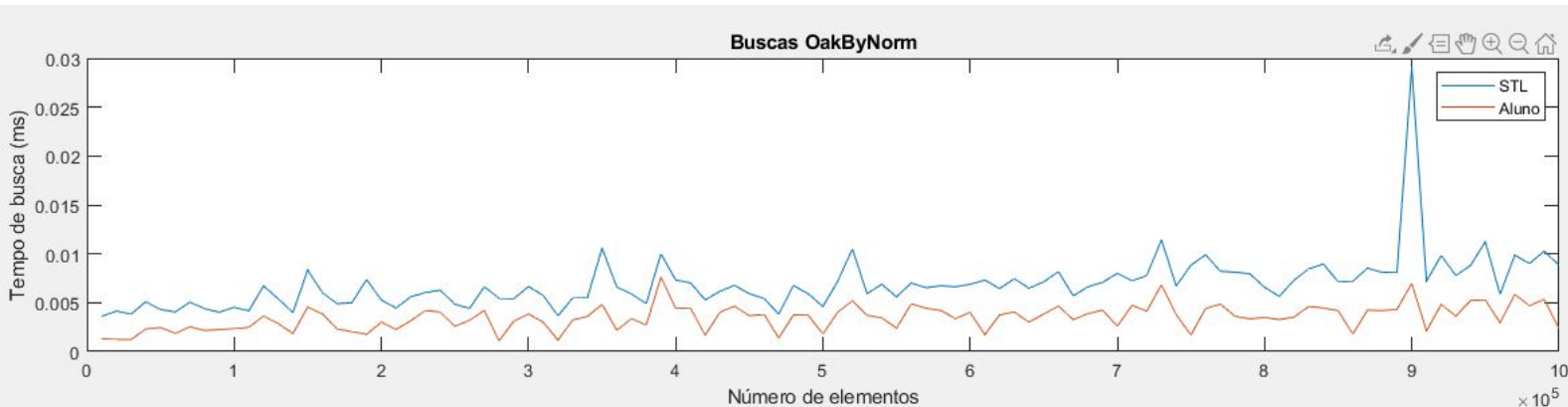
1) Descreva a estrutura que escolheu:

- Qual estrutura (e.g. árvore vermelho-preto)
- (se houver) Qual livro utilizou como base da implementação
 - O pseudo-código que mostrei na aula foi retirado do Corben - que ainda contém o pseudo-código das rotações e inserção, mostra devagar os conceitos básicos como o papel dos nós red-black e dos nodes NIL, e prova a corretude do algoritmo usando as invariâncias do loop.
 - Talvez o próprio Sedgwick tenha escrito uma boa referência também, mas o livro que tenho dele não tem nada. Podem haver outros livros dele.
- Alguma consideração que ajude a entender o seu código (implementou algo diferente do usual? Há classes diferentes que mereçam um diagrama de classes ou uma explicação? Vale desenhos a mão fotografados ou diagramas feitos em software.).

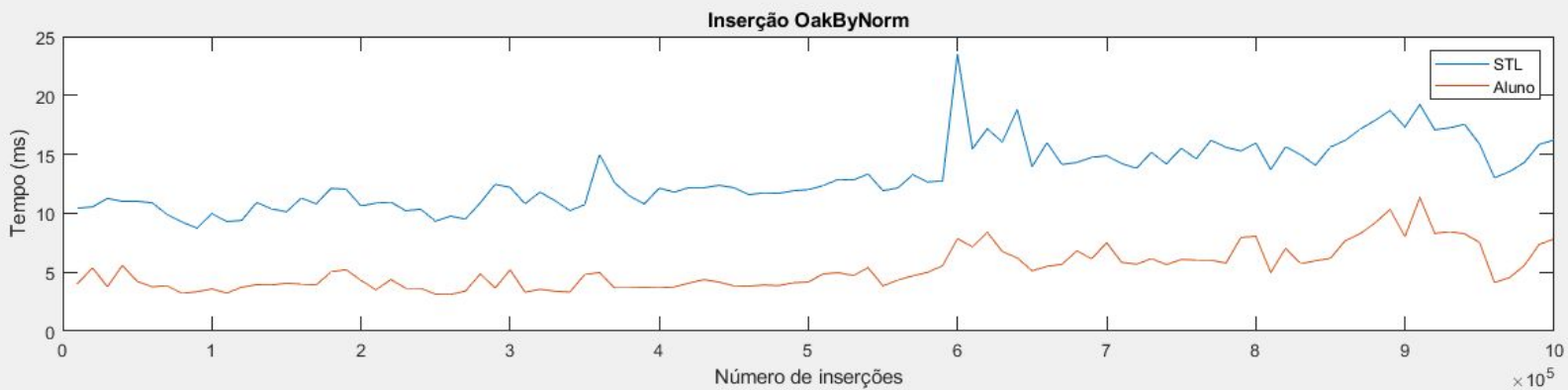
R: Foi utilizada a estrutura de árvore rubro-negra. Existe uma classe de nome RBTree, onde há o ponteiro para o primeiro nó da árvore e onde são realizadas as operações de tamanho, inserção e busca. Também há a classe RBTreeNode, que representa os nós da árvore, onde há as seguintes variáveis: ponteiros para o pai e filhos, booleana que diz se o nó é preto (se não for preto, é vermelho) e objeto 'pair' que guarda a chave e o índice.

- 2) Mostre na forma de gráficos as curvas de tempo do teste OakByNorm, comparando a sua implementação com a minha implementação baseada em `stl::multimap`. Mostre que as buscas e inserções são $O(\log n)$ em ambas as implementações - use um programa de fit em função logarítmica.

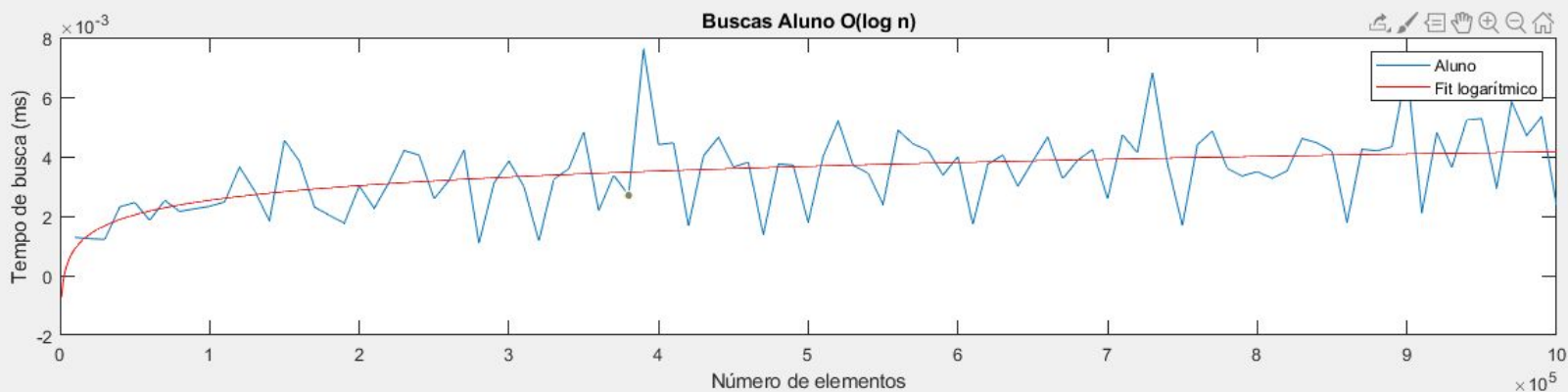
R: Para as operações de busca:



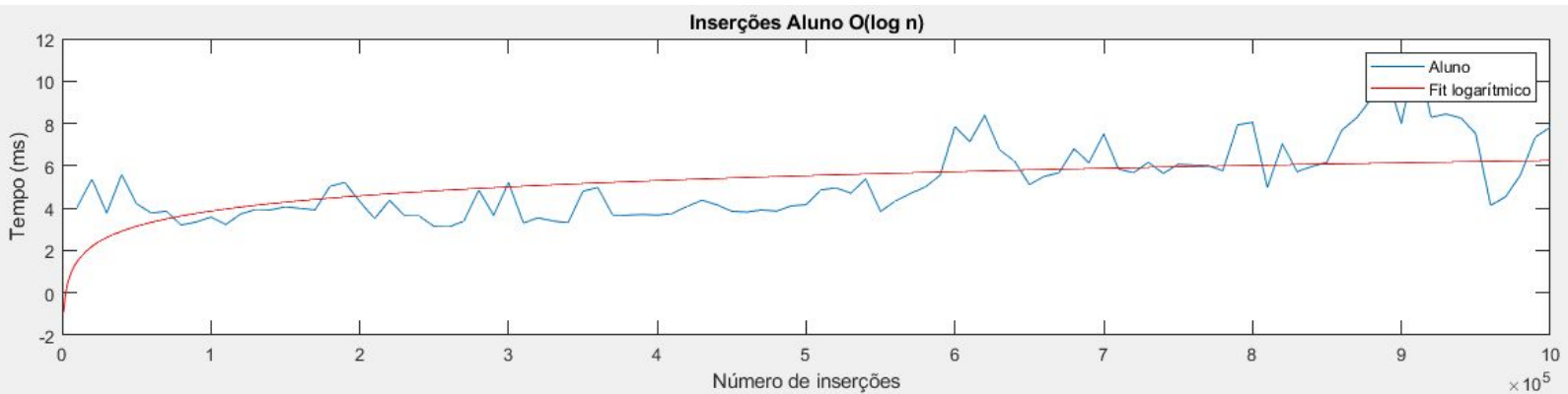
Para as operações de inserção:



Fazendo o *fit* logarítmico para o gráfico de busca na implementação do aluno:



Para o gráfico de inserções:



Nota-se que as operações de inserção e busca são realmente de escala logarítmica.