

Relatório do Laboratório 2 - Balanced Tree

A estrutura utilizada foi uma árvore rubro-negra. Essa é uma árvore de busca binária balanceada que possui em cada um de seus nós o valor da chave, o índice, ponteiros para os filhos e para o pai, além de um bit que pode ser RED ou BLACK. Na implementação, foi criada dentro da classe `IndexPointsAluno` uma *struct* `RedBlack`, que contém os ponteiros para a raiz e para o nulo, além do `long numsize`. Na função construtora de `IndexPointsAluno`, foi inicializada a árvore com a construção dos nós “nulo” e “raiz”. A variável do tipo *struct* `RedBlack` é declarada no `private` da classe `IndexPointsAluno`.

A base para a implementação da árvore foi o livro do Cormen e seu pseudocódigo de árvore rubro-negra. É possível comparar o tempo de inserção e busca para os métodos STL `MultiMap` e a árvore rubro-negra implementada. As comparações são mostradas na figura 1 e na figura 2. Ambas as figuras estão com curvas Log sobrepostas para comparação.

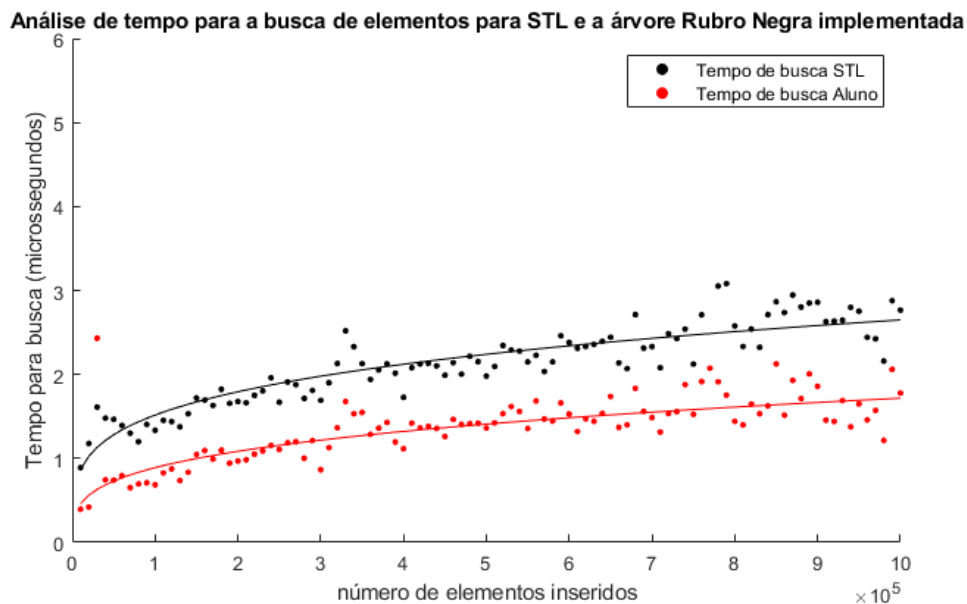


Figura 1: Gráficos do tempo de busca de nós, conforme o número de nós do multimap e da árvore.

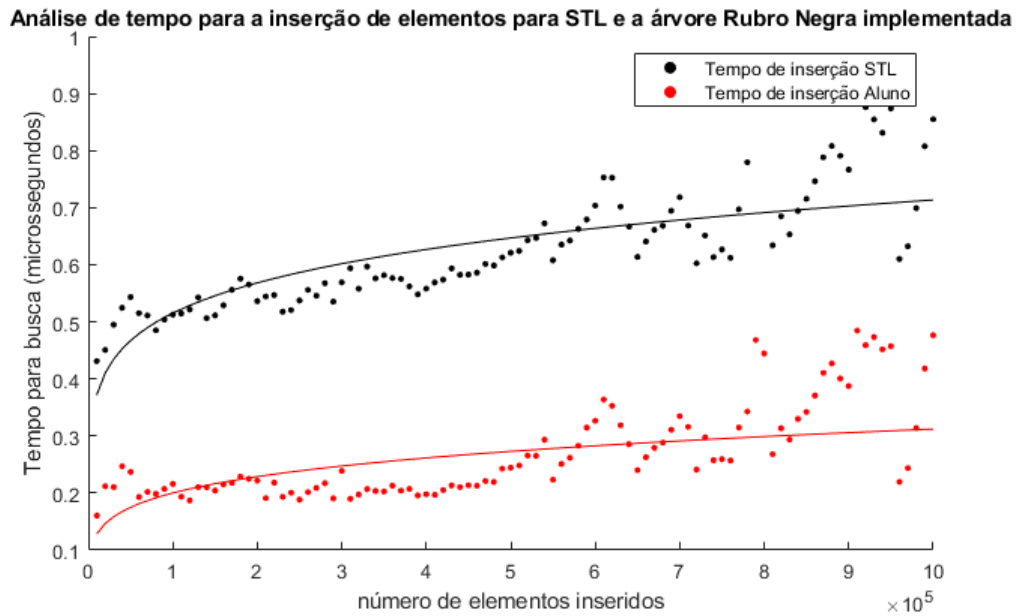


Figura 2: Gráfico do tempo de inserção de nós, conforme o número de nós do multimap ou da árvore.

Nota-se que os pontos, em ambas as figuras 1 e 2, para o Multimap e para a árvore, podem ser aproximados para uma função logarítmica. Isso condiz com a teoria de que árvores de buscas binárias balanceadas, como é o caso da Rubro-Negra, possuem complexidade de tempo $O(\log n)$ para busca e inserções.