

UNIVERSIDADE FEDERAL DO CEARÁ

Alunos: José Robertty e Maria Tassiane
Disciplina: Estrutura de Dados Avançada
Professor: Fábio Dias

Trabalho Final: Estudo computacional das principais árvores balanceadas

1. Introdução

Este trabalho visa expor os resultados dos testes computacionais proposto na disciplina de Estrutura de Dados Avançada. Iremos apresentar os tempos em milissegundos de execuções sobre um determinado conjunto de dados. Teremos em mãos 10391 instancias sobre a classe Empresa. A classe Empresa possui informações de: nome, cnpj, inscrição estadual, data de abertura, site, email, cep, rua, numero, bairro, cidade, estado, telefone, celular. Para este trabalho iremos utilizar como chave a informação “telefone” e implementamos os métodos do Comparable do Java para que possamos construir nossas árvores balanceadas.

A função compareTo consiste em transformar a informação telefone, que inicialmente é salva como uma String, e iremos eliminar os caracteres indesejáveis e transformar a String em um Long. Então cada classe teremos um Long associado e que iremos compará-los para sabermos qual o maior dos dois.

2. Resultados

2.1- Árvore Avl

A árvore Avl é uma famosa árvore balanceada, que consiste em basicamente manter a diferença de alturas da subárvores ser menor que 2, para isso se utiliza de rotações que não entraremos em detalhes por não ser o foco deste trabalho.

Computamos primeiramente o tempo de inserção de todos os 10391 termos, depois computamos o tempo de busca de 30% dos dados, escolhidos aleatoriamente. E depois iremos remover todos os elementos buscados no passo anterior.

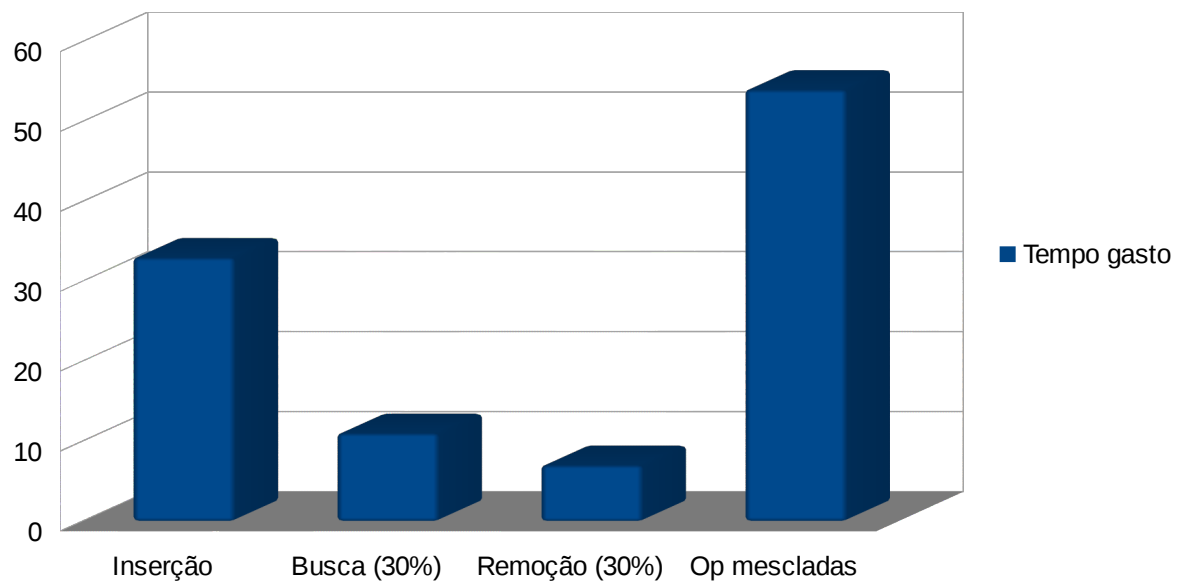
O outro tempo que computamos foi o tempo total onde a cada 20% dos registros inseridos, selecionamos 30% dos que já foram inseridos para realizar a busca e 10% para remoção.

Os resultados dos tempos em milissegundos foram:

Atividade	Tempo gasto
Inserção	33
Busca (30%)	11
Remoção (30%)	7
Operações mescladas	54

Tabela 1 – Tempos das operações sobre a árvore AVL.

Gráfico 1 – Tempos das operações sobre a árvore AVL



2.2- Árvore Splay

A árvore Splay utiliza a ideia de tentar deixar mais próximos da raiz os nós mais acessados.

Computamos primeiramente o tempo de inserção de todos os 10391 termos, depois computamos o tempo de busca de 30% dos dados escolhidos aleatoriamente. E depois iremos remover todos os elementos buscados no passo anterior.

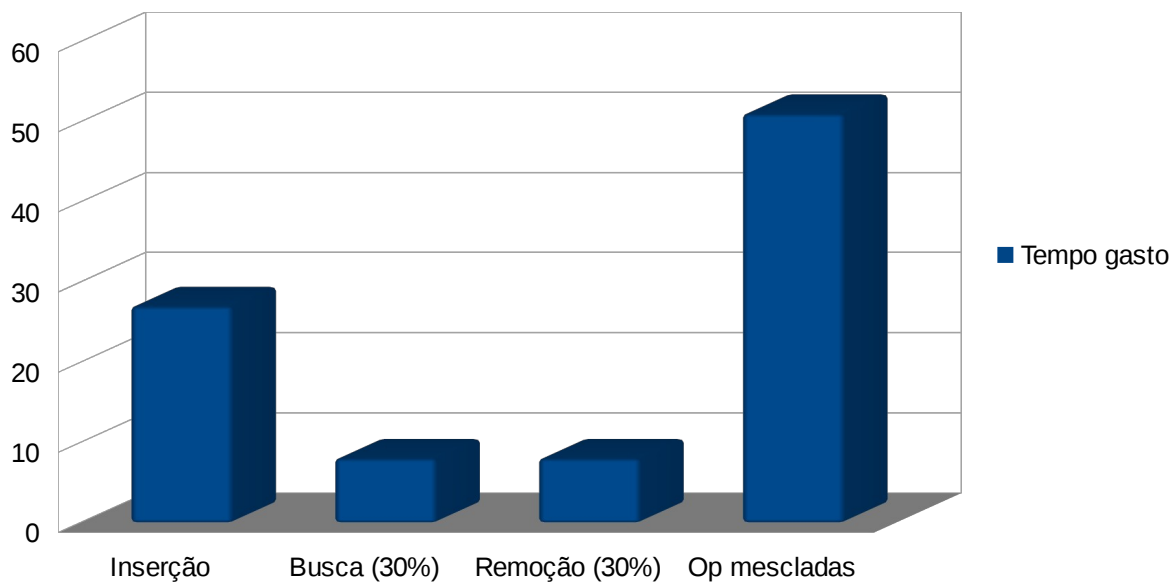
O outro tempo que computamos foi o tempo total onde a cada 20% dos registros inseridos, selecione 30% dos que já foram inseridos para realizar a busca e 10% para remoção.

Os resultados dos tempos em milissegundos foram:

Atividade	Tempo gasto
Inserção	27
Busca (30%)	8
Remoção (30%)	8
Operações mescladas	51

Tabela 2 – Tempos das operações sobre a árvore Splay.

Gráfico 2 – Tempos das operações sobre a árvore Splay



2.3- Árvore Rubro Negra

A árvore Rubro Negra é outra famosa árvore balanceada, em que iremos atribuir cores aos nós, e a partir dessa coloração e de um conjunto de regras (que não iremos explicitar neste trabalho) para que a árvore fique balanceada e as suas operações sejam menos complexas computacionalmente do que a árvore Avl.

Computamos primeiramente o tempo de inserção de todos os 10391 termos, depois computamos o tempo de busca de 30% dos dados escolhidos aleatoriamente. E depois iremos remover todos os elementos buscados no passo anterior.

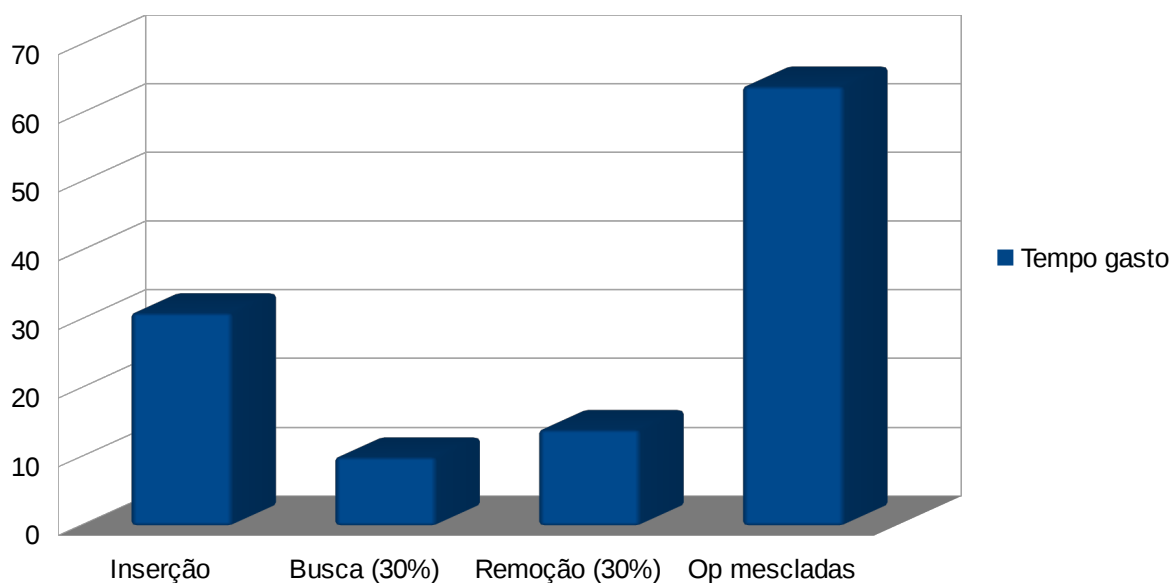
O outro tempo que computamos foi o tempo total onde a cada 20% dos registros inseridos, seleccione 30% dos que já foram inseridos para realizar a busca e 10% para remoção.

Os resultados dos tempos em milissegundos foram:

Atividade	Tempo gasto
Inserção	31
Busca (30%)	10
Remoção (30%)	14
Operações mescladas	64

Tabela 3 – Tempos das operações sobre a árvore Rubro Negra.

Gráfico 3 – Tempos das operações sobre a árvore Rubro Negra



3- Conclusão

Podemos notar que a árvore AVL leva um pouco de vantagem em relação a árvore Rubro Negra, pois a sua remoção possui uma complexidade bem menor que a Rubro Negra. Observamos que a Rubro Negra foi executada em menos tempo nas atividades de Inserção e Busca, porém na atividade de remoção ela se mostra muito inferior a árvore AVL.

A árvore que se mostrou ser mais eficiente foi a árvore Splay. A sua simplicidade de inserção e a atualização constante dos nós mais utilizados para o topo da árvore fazem com que a remoção sejam relativamente rápidas. Na busca acreditamos que os dados tenham interferido no resultado, pois se existir nos dados muitas empresas com a mesma chave (o mesmo telefone, no caso), temos que a busca é realmente muito mais eficiente do que nas demais árvores.

4- Especificações

- Sistema operacional: Ubuntu 15.10
- Processador: Intel core i5-4210U 1.70GHz
- Memória: 7,7 GB
- Tipo de sistema: 64-bit