Instituto de Computação

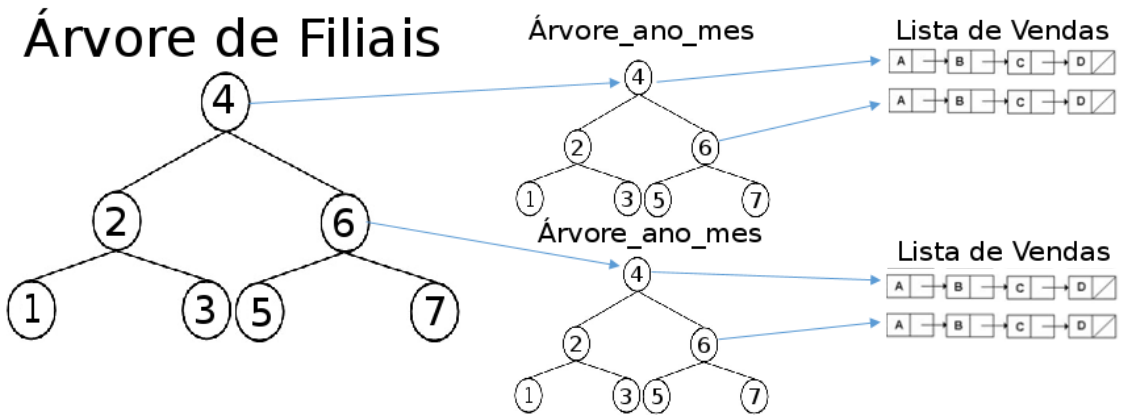
Disciplina de estrutura de dados: Prof. Dr. Luiz André Portes Paes Leme

Aluno: Ricardo George Bhering

Relatório de implementação:

Proposto o problema do trabalho, que tem como objetivo criar um algoritmo que carregue os dados contidos em um arquivo de texto em uma estrutura de dados, permitindo assim responder aos três exemplos de busca apresentados de forma eficiente, escolhi uma estrutura baseada em "árvore de busca binária AVL", composta por uma árvore de árvores e listas encadeadas para compor essa estrutura.

Representação da Estrutura de Dados:



O uso da estrutura composta por árvores AVL, foi escolhida devido a suas características que otimizam as operações de busca, inserção e remoção de elementos e possuem complexidade da ordem de O(log n) no qual n é o número de elementos da árvore.

A primeira árvore de filiais, armazena em cada NÓ de sua estrutura uma chave única, "o número da Filial". Usei como padrão para criação do meu arquivo de dados, o número de 50 filais, relacionando 3 anos consecutivos de vendas e cobrindo de janeiro de 2014 a dezembro de 2017. Todas as empresas possuem vários lançamentos em todos os meses do período citado.

Cada "Nó" dessa primeira árvore de "Filiais", aponta para uma outra árvore com as datas de vendas apelidada de "ano_mes", contendo todos os períodos onde ocorreram vendas desta filial, usando o valor "ano_mes" como parâmetro para a chave de busca.

Esta segunda árvore, armazena todas as datas em que ocorreram lançamentos desta filial, separando cada (ano_mes) em um nó específico. Este NÓ por sua vez, aponta para uma lista de vendas, que contém as informações de códigos dos vendedores e o valor de suas vendas.

Foi utilizado um arquivo chamado "dados.txt", contendo dados de 50 filiais e quatro mil lançamentos para carregar e testar o programa inicialmente. Após o teste de execução das consultas e validação do correto funcionamento dos métodos, um segundo arquivo de dados foi utilizado, o "dadosmil.txt", contendo um milhão de registros gerados aleatoriamente.

O teste de desempenho de carregamento apresentou os seguintes resultados práticos, rodando em um notebook intel core i3, 2365M, dual core com 12GB de memória RAM :

Experimento 1:

O algoritmo executou "carregaestrutura" com o tempo de 3434 milissegundos, realizando a inserção dos dados de 1.000.000 de linhas de registros na estrutura, contendo 3 anos de lançamentos de vendas entre janeiro de 2014 a dezembro de 2017 (36 meses).

Resultados:

Exercício 1: Tempo de busca de todas das filiais (entre 10 a 20), contendo todos os lançamentos de vendas das filiais: O algoritmo executou o "relatorioVendas1" em 30 milissegundos.

Exercício 2: Tempo de busca de todas das filiais (entre 10 a 20), contendo todos os lançamentos de vendas das filiais entre (201701 e 201706): O algoritmo executou o "relatorioVendas2" em 22 milissegundos.

Exercício 3: Tempo de busca de todas das filiais, contendo todos os lançamentos de vendas das filiais entre (201708 e 201710): O algoritmo executou o "relatorioVendas3" em 79 milissegundos.

Comparativamente o arquivo "dadosmil.txt" de 1.000.000 de linhas, é 250 vezes maior que o "dados.txt" e executou as buscas de forma eficiente, mesmo com o enorme crescimento dos lançamentos de vendas.

Experimento 2:

Os resultados do carregamento e busca usando arquivo "dados.txt", com 4000 linhas e lançamentos de 50 filiais e os mesmos 36 meses de lançamentos, realizando um breve comparativo.

Tempo do carregamento de dados na estrutura: O algoritmo executou "carregaestrutura" em 146 milissegundos.

Exercício 1: O algoritmo executou o "relatorioVendas1" em 5 milissegundos. Comparativamente, gastou 25 milissegundos a mais para buscar no arquivo de 1.000.000 de registros.

Exercício 2: O algoritmo executou o "relatorioVendas2" em 12 milissegundos. Comparativamente, gastou 10 milissegundos a mais para buscar no arquivo de 1.000.000 de registros.

Exercício 3: O algoritmo executou o "relatórioVendas3" em 29 milissegundos. Comparativamente, gastou 50 milissegundos a mais para buscar no arquivo de 1.000.000 de registros.

Os testes realizados, mesmo que resumidos, demonstram que a Estrutura de Dados escolhida, consegue atender as buscas de forma eficiente mesmo com o crescimento exponencial dos dados.

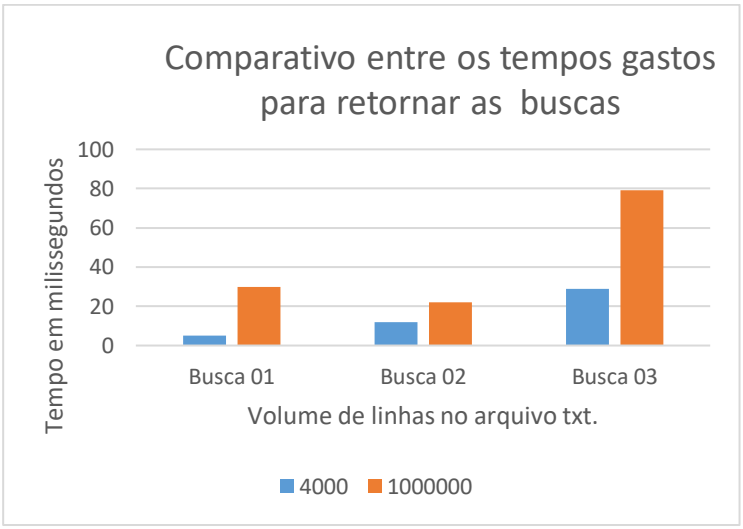


Figura 01: Comparativo entre os tempos gastos para retornar as buscas usando as duas bases de dados.