

Desempenho AVL x Rubro-Negra

O Projeto Final de nossa disciplina de ED2, consistirá na elaboração de um programa para medir experimentalmente o tempo gasto no preenchimento de duas árvores, uma AVL e uma Rubro-Negra, com o objetivo de comparar o desempenho entre as duas. Este preenchimento será efetuado com uma massa de dados coletados de um arquivo, que será fornecido em formato “.CSV”, que consiste em um arquivo de texto puro organizado em formato de tabela, onde suas colunas são delimitadas por um marcador, também chamado de “token”, que neste caso é um ponto e vírgula e o caractere de “nova linha” (\n) ao fim de cada linha.

Tratamento nos dados serão necessários, de acordo com as **suas necessidades**, para que seja possível a inserção dos dados nas duas árvores. Na tabela 1 temos o exemplo de como serão fornecidos os dados a serem inseridos:

Tabela 1 - Massa de Dados desordenada

	1	2	3	4	5	6
1	codigo	nome	idade	empresa	departamento	salario
2	29749	Galena K. Howard	62	Altavista	Research and Development	3297,73
3	44896	Steel Avila	58	Finale	Sales and Marketing	3973,97
4	39153	Leah Moses	45	Macromedia	Public Relations	2356,20
5	3122	Carla Dickson	67	Lycos	Legal Department	1779,99
6	13942	Isaac Reid	36	Lavasoftware	Customer Service	7307,67
7	16728	Kyle A. Head	31	Cakewalk	Public Relations	6989,13
8	5094	Isaiah Y. Gordon	18	Borland	Human Resources	6054,19

Estruturas (structs), pertinentes aos tipos de dados presentes na massa de dados fornecida, serão necessárias para o devido armazenamento nas Árvores AVL e Rubro-Negra. Cada linha presente no arquivo, corresponde à uma estrutura completa. Cada campo desta estrutura está delimitado pelo “token” (;) e o último campo é finalizado com uma quebra de linha, ou nova linha (“\n”), exemplo:

```
93223;Hillary E. Wooten;44;Adobe;Sales and Marketing;2387,7
48171;Raven M. Harrington;40;Yahoo;Tech Support;8142,29
```

Estas estruturas terão que estar obrigatoriamente dentro dos nós das árvores.

Nesta etapa estarão presentes as medições de tempo gasto para a ordenação e o armazenamento, sendo tomado o tempo desde o momento da aquisição dos dados no arquivo, e até que as árvores estejam totalmente preenchidas.

Duas medições serão realizadas para cada árvore, para efeitos comparativos:

1. A massa de dados desordenada presente no arquivo fornecido alimentará diretamente as duas árvores, e o tempo de duração do processo será medido;
2. Uma segunda massa de dados ordenados gerada por seu programa, descrita mais adiante, alimentará os dois tipos de árvores e o tempo de duração do processo será medido.

As tomadas de tempo devem acontecer desde o momento da leitura dos dados no arquivo, até a inserção da última estrutura nas árvores. Exclui-se da tomada de tempo o processo de ordenação em vetor auxiliar e a geração do arquivo ordenado.

A forma de preenchimento das árvores, com os dados do arquivo desordenado ou dados do arquivo ordenado, deverá ser escolhida em um menu, portanto um bom procedimento na inicialização do programa, antes de qualquer inserção ou mesmo a criação das árvores, seria a ordenação dos dados e geração do segundo arquivo já ordenado.

Geração do segundo arquivo

A partir da massa de dados desordenada, fornecida no arquivo .CSV, deverá ser gerado um segundo arquivo com o mesmo conteúdo e do mesmo tipo, porém ordenado. Um algoritmo de ordenação deverá ser utilizado para a tarefa e também será necessária a utilização de um vetor auxiliar alocado com a mesma quantidade de elementos presentes no arquivo fornecido. Os dados então devem ser transferidos do arquivo fornecido, para o vetor auxiliar alocado, e este então passado ao algoritmo de ordenação.

Os dados após a ordenação devem ser salvos em um arquivo de texto no mesmo formato do arquivo original que contém os dados desordenados, para garantir a **igualdade de condições**. A partir deste ponto, a leitura e inserção nas árvores acontece de forma igual, tanto para o arquivo ordenado, quanto o desordenado, portanto, é completamente viável o reaproveitamento do código utilizado na etapa 1.

A ordenação do arquivo de dados deverá ser **obrigatoriamente realizada internamente por seu programa**. Ordenações externas (Ex.: Excel), **não serão aceitas**. A escolha do algoritmo de ordenação ficará a critério do grupo, levando-se em consideração os resultados obtidos no primeiro projeto avaliativo, ou seja, o algoritmo que obteve melhor desempenho em seus testes, na etapa P1.

Para efeitos comparativos, seu programa deve exibir os resultados para a inserção de dados ordenados e desordenados para os dois tipos de árvores, ou seja, resultado para inserção direta sem ordenação dos dados presentes no arquivo original fornecido, e o resultado para inserção dos dados ordenados presentes no arquivo secundário gerado por sua aplicação. Os tempos em todos os casos devem ser tomados no início da leitura dos dados nos arquivos e até o término da inserção nas árvores.

Relatório conclusivo

Uma vez que o programa esteja implementado e funcionando, um relatório em padrão ABNT deverá ser elaborado, indicando os resultados comparativos dos testes: qual tipo de árvore teve melhor desempenho para o preenchimento, qual a melhor forma de se preencher, se previamente ordenada ou totalmente desordenada, deixando a cargo das funções internas das árvores esse trabalho.

O relatório deverá apresentar os resultados para os dois tipos de inserção de elementos nas duas árvores, previamente ordenados e totalmente desordenados.

A massa de dados fornecida conta com 15.000 elementos.

Para os resultados apresentados é interessante e desejável a utilização de um gráfico de performance para efeitos de comparação visual.

Condições e prazo de entrega

O projeto poderá ser realizado de forma **individual** ou em grupo, neste caso mantendo os grupos já formados em trabalhos anteriores.

A entrega será efetuada via plataforma Moodle com **todos os arquivos pertencentes ao projeto (todos os arquivos gerados pelo CodeBlocks), inclusive o arquivo secundário de dados ordenados gerado**, em data definida na mesma.

Seu projeto será avaliado pela qualidade da implementação (organização e modularização do código), pelo cumprimento de todos os requisitos e pela apresentação dos resultados no relatório final.

Trocas de ideias e discussões com outros participantes são válidas e construtivas, mas você deve ser o responsável por toda a codificação do projeto. Será atribuída **NOTA ZERO** a projetos de código idêntico ou mesmo tenha sido “**maquiado**”, tanto o copiado quanto o projeto que serviu de fonte, **principalmente**.