

Trabalho Prático 03

Benchmark de Pesquisa

Faça um programa que leia um arquivo texto (no formato `.txt`) e imprima, em ordem alfabética, as palavras e a suas frequências no texto, além de indicadores de performance dos algoritmos implementados.

A leitura do arquivo deverá desprezar espaços em branco e sinais de pontuação, que serão considerados separadores de palavras. Além disso, a leitura deverá converter todas as letras maiúsculas em minúsculas.

A pesquisa e inserção das palavras do texto deverão ser implementadas com as seguintes estruturas:

1. Pesquisa Binária (utilizando um vetor dinâmico para armazenar as palavras).
2. Árvore Binária de Pesquisa sem balanceamento.
3. Árvore Binária de Pesquisa com balanceamento (Árvore AVL).

Coloque contadores no seu programa para determinar o número de comparações de chaves e atribuições de registros necessárias para montar a tabela de frequências em cada uma das estruturas acima. Conte apenas o número de comparações para montar a estrutura (operações de inserir e pesquisar). Você não deve considerar as operações na fase de impressão ordenada.

Calcule também o tempo que cada estrutura leva para montar a tabela. Analise, por meio dos dados coletados, a eficiência de cada estrutura.

Entrada

A entrada de dados será um arquivo de texto contendo um texto qualquer. O arquivo texto deverá usar a codificação UTF-8. Seu algoritmo receberá dois parâmetros: (i) nome do arquivo texto a ser lido e (ii) estrutura a ser utilizada para impressão das frequências.

Exemplos de execução:

```
./tp3 entrada.txt pesquisa_binaria
```

```
./tp3 entrada.txt arvore_binaria
```

```
./tp3 entrada.txt arvore_avl
```

Exemplo de arquivo texto:

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Saída:

Deverá ser impresso na tela um resumo completo da execução para todas as três estruturas implementadas. As estruturas devem ser impressas em ordem de desempenho (as que gastaram menos tempo primeiro). Em seguida, as frequências das palavras devem ser impressas. Para esta impressão, deverá ser utilizada a estrutura indicada pelo parâmetro passado na linha de comando. Segue um exemplo **fictício** para indicar o formato da saída:

```
arvore avl:
    53 comparações
    0.99 segundos

arvore binaria:
    103 comparações
    1.19 segundos

pesquisa binaria:
    103 comparações
    1.89 segundos

frequencia das palavras:
    in: 10
    ipsum: 5
    lorem: 3
    ut: 20
```

Comentários Gerais:

- Clareza, indentação e comentários no código também vão valer pontos. Por isso, escolha cuidadosamente o nome das variáveis e torne o código o mais legível possível.
- Trabalhos copiados (e FONTE) terão nota zero, além de os alunos envolvidos no plágio perderem toda a nota atribuída a participação e pontos extras, entre outros.
- ***O trabalho pode ser feito em dupla***, mas o entrevistado na apresentação será sorteado. Assim, é necessário que ambos tenham conhecimento do trabalho por completo e participem ativamente da sua implementação e documentação.

O que deve ser entregue:

- Código fonte do programa em C ou C++ (bem indentado e comentado).
- Documentação do trabalho. A documentação deve conter:
 1. *Implementação*: descrição sobre a implementação do programa. Não faça “*print screens*” de telas. Ao contrário, procure resumir ao máximo a documentação, fazendo referência ao que julgar mais relevante. É importante, no entanto, que seja descrito o funcionamento das principais funções e procedimentos utilizados, bem como decisões tomadas relativas aos casos e detalhes de especificação que porventura estejam omissos no enunciado. Muito importante: os códigos utilizados na implementação devem ser inseridos na documentação.
 2. *Análise*: deve ser feita uma análise dos resultados obtidos com este trabalho.
 3. *Conclusão*: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
 - 4. *Formato*: PDF.

Como deve ser feita a entrega e apresentação

Verifique se seu programa compila e executa na linha de comando antes de efetuar a entrega.

Quando o resultado for correto, entregue via , até as **23h55 do dia 13/03/2016**, um arquivos .ZIP e um arquivo .PDF com o nome e sobrenome dos dois alunos da dupla. É obrigatório que os dois alunos de cada dupla façam a entrega no *Moodle*.

Exemplo:

- fulano-tal_beltrano-ciclano.zip – este arquivo .ZIP deve conter apenas os arquivos .c (ou .cpp) e .h utilizados na implementação.
- fulano-tal_beltrano-ciclano .pdf – este arquivo .PDF deve conter a documentação do trabalho.

Cada dupla deverá também apresentar seu trabalho, breve explicação e exibição do programa em execução, no dia **14 de março na sala 46 do DECOM** (horários disponíveis: 10h-12h e das 13h-17h). Para a apresentação, não é preciso agendar horário, prevalecendo a ordem de chegada das duplas.