

Estrutura de Dados

Prof. Frederico S. Oliveira - Prazo final 11/04/2016

Segundo Semestre de 2016

Terceiro Trabalho Prático

1 Introdução

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 pesquisa e inserção das palavras do texto deverão ser implementadas com Árvore Binária de Pesquisa e Árvore AVL.

2 Entrada

A entrada de dados será um arquivo de texto contendo um texto qualquer. O arquivo texto deverá usar a codificação UTF-8. A seguir, um exemplo de arquivo de entrada:

```
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.
```

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. Você pode considerar que cada palavra contém no máximo 20 letras e que não haverão caracteres acentuados.

3 Saída

Deverá ser impresso na tela um resumo completo da execução para as duas estruturas implementadas. As estruturas devem ser impressas em ordem de desempenho (a que gastar menos tempo primeiro). Em seguida, as frequências das palavras devem ser impressas. Segue um exemplo fictício para indicar o formato da saída:

```
arvore avl:  
  53 comparações  
  0.99 segundos  
arvore binaria:  
  103 comparações  
  1.19 segundos  
frequencia das palavras:  
  ad 1  
  adipisicing 1
```

```
aliqua. 1
aliquip 1
amet,1
aute 1
cillum 1
commodo 1
consectetur 1
consequat. 1
do 1
dolor 2
dolore 2
```

4 Desenvolvimento

Você deverá implementar uma Árvore de Pesquisa sem Balanceamento e uma Árvore de Pesquisa com Balanceamento (AVL), onde cada nó irá armazenar uma palavra e sua frequência no texto. A medida que as palavras são lidas, o seu algoritmo deve pesquisar na Árvore para ver se a palavra já está presente. Se estiver, incrementa a frequência dessa palavra. Se não estiver presente, cria um novo nó na Árvore. O uso da Árvore de pesquisa é o mais indicado pois a sua estrutura será bastante dinâmica e no final você precisará imprimir as palavras em ordem alfabética.

Para executar seu programa, como parâmetro seu programa deve receber o nome do arquivo de entrada. Por exemplo, se o seu programa chama-se trabalho.o e você quiser enviar o arquivo de entrada “entrada.txt”, sua chamada deve ser:

```
\.trabalho.o entrada.txt
```

5 Documentação

Obrigatoriamente a documentação deve conter:

1. Introdução: descrição do problema a ser resolvido e visão geral sobre o funcionamento do programa.
2. Desenvolvimento: descrição sobre a implementação do programa. Deve ser detalhada a estrutura de dados utilizada (preferencialmente com diagramas ilustrativos), o funcionamento das principais funções e procedimentos utilizados, o formato de entrada e saída de dados, bem como decisões tomadas relativas aos casos e detalhes de especificação que por ventura estejam omissos no enunciado.
3. Compilação e Execução: descrição de como como compilar o código-fonte, bem como a sua execução.
4. Testes: descrição de quais testes foram executados e as saídas obtidas.
5. Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
6. Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo livros e/ou sites da internet se for o caso, seguindo padrão ABNT.

6 Material a ser Entregue

Os seguintes itens devem ser enviados via Ambiente Virtual de Aprendizagem (AVA), em um único arquivo compactado, com extensão “.tar.gz”:

- Código fonte do programa em C, bem indentado e comentado. O código-fonte, bem como os testes de sua execução, valem 50% da nota.
- Documentação do trabalho, que deve ser entregue mandatoriamente no formato PDF. A documentação vale 50% da nota.
- Caso também seja entregue a documentação em LaTeX, ganha-se 1 ponto, em um total de 10 pontos, não acumulativo.

7 Conclusão

Para cálculo da nota, serão realizados testes, e a nota será calculada com base nestes testes. Será atribuído zero aos seguintes casos:

- Código-fonte não compilar no Linux.
- Nenhum teste for executado com sucesso.
- Não apresentar documentação.
- Plágio.

Este trabalho deve ser desenvolvido de forma individual ou em dupla. Não serão realizadas exceções. Bom trabalho.