



UFOP

UNIVERSIDADE FEDERAL DE OURO PRETO  
INSTITUTO DE CIÊNCIAS EXATAS E APLICADAS  
DEPARTAMENTO DE CIÊNCIAS EXATAS E APLICADAS  
ALGORITMOS E ESTRUTURA DE DADOS II

Aluno:

3º Trabalho

Professor: Pollyanna Dias

Data de entrega:

Curso: EC/SI

Turno:  
Tarde/Noite

Período: 3º

Valor: 15 Nota:

**Trabalho Prático 2**  
**Algoritmos de Pesquisa/Busca**

Faça um programa que leia um texto qualquer (arquivo no formato *.txt*) e imprima, em ordem alfabética, as palavras e a sua frequência no texto. Por exemplo, no texto *“dois mais dois são quatro”* o seu programa deverá imprimir:

**dois 2**  
**mais 1**  
**quatro 1**  
**são 1**

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 existem no máximo 1024 palavras diferentes no texto (de fato, considere as primeiras 1024 diferentes palavras que aparecerem no texto e despreze as seguintes novas palavras), e que cada palavra contém no máximo 20 letras.

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

1. Pesquisa Seqüencial.
2. Pesquisa Binária.
3. Árvore Binária de Pesquisa sem balanceamento.
4. Árvore Binária de Pesquisa com balanceamento.
5. Hash – endereçamento aberto.

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 (Apenas o número de comparações para montar a estrutura. Não é necessário considerar as comparações e atribuições para a impressão ordenada).

Calcule também o tempo que cada estrutura leva para montar a tabela. Análise através dos dados coletados o desempenho / eficiência de cada estrutura. Use parâmetros de linha de comando para fazer sua chamada. Esse tipo de execução é bastante comum em sistemas Unix / Linux e no antigo DOS. Por exemplo, se o seu programa chama-se Freq e você quiser contar a frequência do arquivo texto.txt utilizando um hash, sua chamada deve ser:

```
> Freq hash texto.txt
```

Descubra como ler os parâmetros da linha de comando (da mesma forma como foi feita no trabalho prático 3) e defina (e documente) a sintaxe a ser utilizada. Também faz parte do trabalho descobrir como fazer a leitura de um arquivo texto e a manipulação adequada de strings.

### O que deve ser entregue

- Código fonte do programa em C ou C++ (bem identada e comentada).
- Documentação do trabalho. Entre outras coisas, a documentação deve conter:
  1. Introdução: descrição do problema a ser resolvido e visão geral sobre o funcionamento do programa.
  2. Implementação: descrição sobre a implementação do programa. Deve ser detalhada a estrutura de dados utilizada (de preferência 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 porventura estejam omissos no enunciado.
  3. Estudo de Complexidade: estudo da complexidade do tempo de execução dos procedimentos implementados e do programa como um todo (notação O).
  4. Listagem de testes executados: os testes executados devem ser simplesmente apresentados.
  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 sites da Internet se for o caso.
  7. Formato: mandatoriamente em PDF

Obs1: Consulte as dicas do Prof. Nívio Ziviani de como deve ser feita uma boa implementação e documentação de um trabalho prático:  
<http://www.dcc.ufmg.br/~nivio/cursos/aed2/roteiro/>

### Como deve ser feita a entrega:

A entrega DEVE ser feita pelo email: pollyanna.dias@decea.ufop.br na forma de um **único** arquivo zipado, contendo o código, os arquivos e a documentação. Também deve ser entregue a documentação impressa na próxima aula (teórica ou prática) após a data de entrega do trabalho.

- Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar;
- Clareza, identificação e comentários no programa também vão valer pontos;
- O trabalho pode ser feito individualmente ou em grupo de dois alunos;
- Trabalhos copiados (e FONTE) terão nota zero;
- Trabalhos entregue em atraso serão aceitos, todavia a nota atribuída ao trabalho será zero
- Evite discussões inócuas com o professor em tentar postergar a data de entrega do referido trabalho.