

## Trabalho Prático 2

### Indexador de Arquivos

valor: 25 pontos

Data de entrega: 10/07/2019

#### 1. Objetivo:

O objetivo deste trabalho é utilizar analisar o desempenho de diversas estruturas de dados para indexar o conteúdo de arquivos.

#### 2. Descrição

A indexação de arquivos é uma tarefa importante para buscar conteúdo dentro dos arquivos de sistemas. Neste trabalho, consideraremos que todos os arquivos cabem na memória principal. Para indexar o conteúdo de vários arquivos, é necessário carregar todo o conteúdo do arquivo, marcar onde ocorrem as ocorrências para cada palavra.

Por exemplo, considere que queremos indexar os arquivos abaixo:

lorem.txt
<code>Lorem ipsum dolor sit amet, consectetur adipiscing elit ipsum.</code>

Ipsum.txt
<code>ullamco laboris nisi ut aliquip ex ea commodo consequat. Lorem ipsum dolor sit amet.</code>

Ao pesquisar a palavra *ipsum*, o resultado será o nome do arquivo seguido pelas posições em que a palavra ocorre.

<i>Saída</i>
<i>lorem.txt 7, 58</i>
<i>Ipsum.txt 64</i>

Para este trabalho, deverão ser comparadas versões do indexador com diferentes tipos abstratos de dados.

### 3. Sua Função (Requisitos):

Avaliar o tempo para construção da estrutura de dados, carregando todos os arquivos na memória; avaliar o tempo para busca de palavras (aleatórias nas estruturas); liberar a memória completamente.

O programa deverá receber como argumento os nomes dos arquivos que serão carregados e o número de palavras aleatórias que serão pesquisadas. Além disso, deve ser possível configurar ou informar por parâmetros quais estruturas de dados serão avaliadas.

As seguintes estruturas de dados devem estar disponíveis no programa:

- Listas
- Árvore binária não balanceada
- Árvore binária balanceada (AVL)
- Árvores de prefixo (TRIE)
- Tabela Hash

Seu programa deverá ter uma função chamada AvaliaDesempenho(int N, void \*Estrutura) que carrega todas as palavras, seleciona N palavras entre as carregadas aleatoriamente, busca essas palavras na Estrutura e gera como saída um relatório de desempenho.

Exemplo fictício da saída de um programa:

N	NomeEstrutura	TempodeBusca
10	Lista	1
1000	Lista	10
10	ABB	1
1000	ABB	5
1000	AB Bal.	3
1000	Hash	3

Você pode mudar/organizar essa saída para o que achar mais conveniente em seu relatório.

Cabe a você desenvolver os TAD e a forma de armazenamento das informações. Ressalta-se que essa é a parte principal do trabalho, e cópias serão severamente punidas.

O seu programa deverá funcionar via linha de comando.

Teste com os mais variados tamanhos de arquivos texto sem formatação (txt) e com diversos valores de N.

Os arquivos devem variar de 1 Kilobytes até o limite do seu PC. Importante que você chegue até os limites do seu computador. Por exemplo, carregar 4Gb ou mais de arquivos textos.

### 3. O que deve ser entregue:

- **(59%) Código fonte** identado e bem comentado, com bibliotecas bem definidas.
- (1%) README.TXT - Arquivo texto com os comandos necessários para compilar e executar o programa. **Preferencialmente o MAKEFILE.**
- **(40%) Relatório contendo:**
  - Introdução: descrição do problema a ser resolvido e visão geral sobre o funcionamento do programa.
  - Implementação: descrição sobre a implementação dos algoritmos, incluindo a estrutura de dados que armazena os índices; Deve ser detalhada a estrutura de dados utilizada (de preferência com diagramas ilustrativos), o funcionamento dos algoritmos, e decisões tomadas relativas aos detalhes de especificação que porventura estejam omissos no enunciado.
  - Análise dos testes: analisar os tempos para carregar e para fazer buscas de N elementos para arquivos de tamanhos diversos.
  - Utilizar gráficos para ilustrar o desempenho da sua implementação.
  - Conclusão: comentários gerais sobre quais algoritmos/TADs são os mais eficientes.
  - Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites da Internet se for o caso. Uma referência bibliográfica deve ser citada no texto quando da sua utilização
  - Em Latex: Caso o trabalho seja elaborado/escrito em latex, ganha-se 0,1 pontos.
  - Formato: deve ser PDF

**Dica:** Um modelo do trabalho em Latex está disponível em:

<https://www.overleaf.com/read/tdhmvmhkrfsv>

Fazer o download do .zip (Clique em Menu --> Source) e depois criar seu próprio projeto no Overleaf, ou em outro editor, usando esse mesmo .zip.