

## Lista de Exercícios

### Árvore AVL

#### Questão única:

Faça um programa que leia um arquivo texto (em *.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 dos registros necessários 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.

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 que deve ser lido e (ii) estrutura a ser utilizada para impressão das frequências.

Exemplos de execução:

```
./tp entrada.txt pesquisa_binaria
```

```
./tp entrada.txt arvore_binaria
```

```
./tp entrada.txt arvore_avl
```

Exemplo de arquivo texto:

```
Lorem ipsum dolor sit amet, consectetur adipisicing 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
```

Deverá ser impresso na tela um resumo completo da execução para todas as três estruturas implementadas. O tempo gasto por cada estrutura. 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 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
```

### Tabelas de Dispersão

- 1- Observando cada um dos caracteres que formam a palavra P E S Q U I S A. Crie uma tabela de dispersão (hash table) que armazene cada um destes caracteres, utilizando a seguinte função:

$$h(\text{char}) = (\text{ordem de char no alfabeto}) \% 7$$

Desenhe o esquema de como ficaram a hash table e as listas encadeadas.

- 2- Implemente uma classe Hash respeitando o seguinte:
  - a) Utilize uma estrutura de tabela Hash para armazenar os elementos que devem ser inseridos e consultados com complexidade  $O(1)$ .

- b) A estrutura criada deve permitir o armazenamento de elementos de qualquer tipo. Elementos esses identificados por uma chave do tipo string.

### **Introdução à Teoria Computacional dos Grafos**

1 – Faça os itens a seguir:

- a) Desenhe um grafo que tenha o conjunto de vértice  $V=\{1,2,3,4,5\}$ , o conjunto de arestas  $E=\{a1,a2,a3,a4,a5,a6\}$  e a função  $g(a1)=\{1,2\}$ ,  $g(a2)=\{1,3\}$ ,  $g(a3)=\{3,4\}$ ,  $g(a4)=\{3,4\}$ ,  $g(a5)=\{4,5\}$  e  $g(a6)=\{5,5\}$ .
- b) Encontre no grafo do item (a): i) dois vértices não adjacentes; ii) um laço; iii) o grau do vértice 3
- c) Encontre um isomorfismo do grafo apresentado no item (a)
- d) Apresente a matriz e lista de adjacência do grafo do item (a)

2 – Um grafo  $G$  é planar se existe uma representação (desenho, imersão) de  $G$  no plano de modo que as áreas se encontrem somente nos vértices, isto é, de modo que as arestas não se cruzem. Um grafo planar divide o plano em regiões chamadas faces. Existe sempre uma única face chamada externa ou infinita, que não está limitada (i.e, tem área infinita). Demonstre a Fórmula de Euler: Em um grafo conexo planar com  $f$  faces,  $n$  vértices e  $m$  arestas, vale  $f=m-n+2$ .

3 – Explique o que são grafos eulerianos e hamiltonianos. Dê exemplos e propriedades de cada um.

4 – Quando trabalhamos com grafos admitimos que algumas operações básicas já estejam implementadas. Então nossa primeira tarefa é implementar as seguintes operações básicas tanto para uma representação por matriz de adjacências quanto para lista de adjacências:

- a) Definição do Grafo
- b) Adição de Arestas
- c) Determinação de vizinhos de um dado vértice
- d) Teste de vizinhança entre par de vértices
- e) Remoção de arestas

5 – O processo de busca é comum em diversas áreas de computação. Um algoritmo clássico para realizar buscas em estruturas de dados representadas por grafos é o chamado Algoritmo de Busca em Profundidade. Implemente tal algoritmo.