



Disciplina: Estrutura de Dados I
Professor: Ivairton M. Santos
Ano letivo: 2017/2

Trabalho 8 - Final

Problema 1 - Utilizando a estrutura de dados Pilha, implemente um algoritmo que identifica se uma frase é palíndrome. Adicionalmente o algoritmo deve ser capaz de verificar frases palíndrome com caracteres coringas. Considere os exemplos abaixo:

Para a entrada:

Soa como caos
S** com* *aos
So* c*m* c*os
Soa como cao

Tem como saída:

É palíndrome
É palíndrome - Letras: o, a, o, c
Indefinido (Impossível determinar letras)
Não é palíndrome

Verifique o algoritmo com as entradas:

- 1) A b*s* do te*o desa*a
- 2) *ezo ***do par* **der co**r
- 3) S*co ** r**va coloco no colo cavia* e d**e*
- 4) ** ** ***** ** Laura mil e dez
- 5) Socorram-me subi no ***** ** *****
- 6) R*la *om *al*r
- 7) A *a*a *a*a
- 8) A bo** da lob*
- 9) A ***** da gorda
- 10) *** o p*ema
- 11) O *o*a*o *c*t* a*o*es a damas amada* * r*m* a*a*a * n*m*r*

Problema 2 - Utilizando a estrutura de Pilha, escreva um programa que, dada uma entrada formada por uma frase, avalia se o “balanço” entre os parênteses, chaves e colchetes está correto.

Exemplos de entradas e saídas:

- **Entrada 1:**
O cachorro (preto) muito bonito {e grande} saiu correndo [feito louco] pela rua!
- ✓ **Saída 1:**
Balanceamento de Parênteses=OK, Chaves=OK, Colchetes=OK.
- **Entrada 2:**
O foguete [Apolo], viajou milhares de kilometros (km e se perdeu {{no espaço}}).
- ✓ **Saída 2:**
Balanceamento de Parênteses=ERRADO, Chaves=OK, Colchetes=OK.
- **Entrada 3:**
Corram! Que eles)aqueles maus(estão chegando!
- ✓ **Saída 3:**
Balanceamento de Parênteses=ERRADO, Chaves = (ND), Colchetes = (ND).

Problema 3 - Uma vez por ano a UFMT promove um evento de integração entre alunos da Universidade (com idade ≥ 17) e alunos do ensino fundamental (idade < 17). Neste evento a Universidade transporta, em um ônibus, os estudantes para um passeio. Para promover interação entre os estudantes e evitar bagunça dos mais jovens durante a viagem, decidiu-se colocar o mais velho dos universitários com o mais novo dos fundamentalistas, o segundo mais velho com o segundo mais novo e assim por diante.

Implemente um sistema que leia, de um arquivo de entrada, o nome e as idades dos estudantes, ordene esses nomes, usando um método de ordenação com complexidade melhor que $O(n^2)$ e que ao final gere um arquivo de saída com os nomes dos pares de alunos (e suas idades) que sentarão juntos no ônibus.

Problema 4 - Presuma um arquivo de entrada que contém números inteiros aleatórios com valores no intervalo entre a e b . Este arquivo pode conter vários números repetidos. Para este contexto pode ser implementada uma **Classificação por distribuição**, que ocorre da seguinte maneira: A partir de um vetor, de tamanho $(b-a+1)$, que armazena os valores, defina um vetor `numero[i-a]` como o número de vezes que o inteiro i aparece no arquivo. Escreva uma rotina para ler os valores de um arquivo de entrada (conforme especificado acima) e classificar os valores por meio deste método. Escreva o resultado em um arquivo de saída.

Problema 5 - Centroide é uma representação de um texto que verifica todas as palavras que o compõe. Cada ocorrência de uma palavra deve ser contabilizada. Finalmente, as palavras devem ser organizadas no centroide de maneira ordenada.

Por exemplo, considere o texto: "Sabia que o sabiá sabia assoviar?". Perceba que a palavra "sabia" ocorre 2 vezes, enquanto que as demais ocorrem 1 vez cada. Portanto o centroide correspondente a esta frase deve ser: `[que (1x); o (1x); sabiá (1x); assoviar (1x); sabia (2x)]`

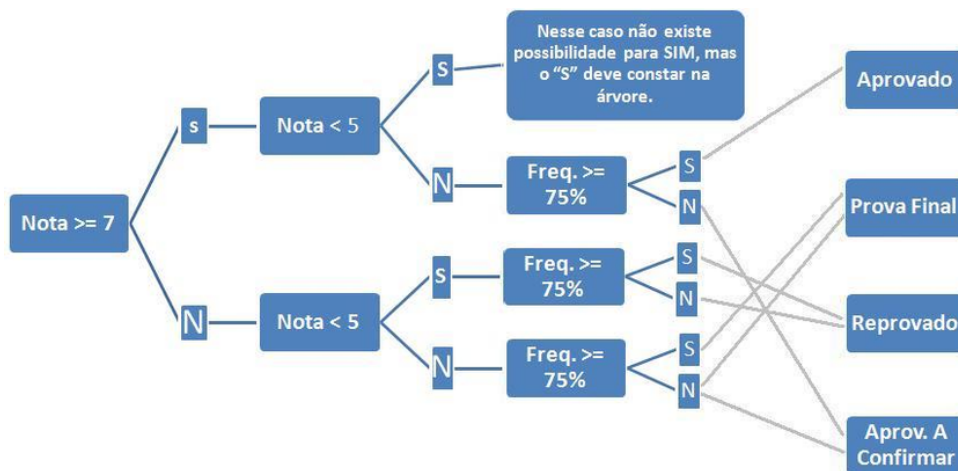
Considere um diretório contendo um ou vários arquivos do tipo texto. Implemente um sistema, utilizando estruturas de dados apropriadas para representar concisamente os arquivos deste diretório, segundo o conceito de centroide (deve ser definida uma estrutura com todos os centroides do diretório e cada arquivo possui um centroide correspondente). Ao final, gere um arquivo de saída que contenha o centroide de cada arquivo e o centroide do diretório.

Pontuação e letras maiúsculas/minúsculas devem ser ignoradas.

Problema 6 - Implemente uma árvore com múltiplos filhos com capacidade $T > 1$ que armazene números inteiros positivos gerados aleatoriamente. Implemente para esta árvore as seguintes funcionalidades:

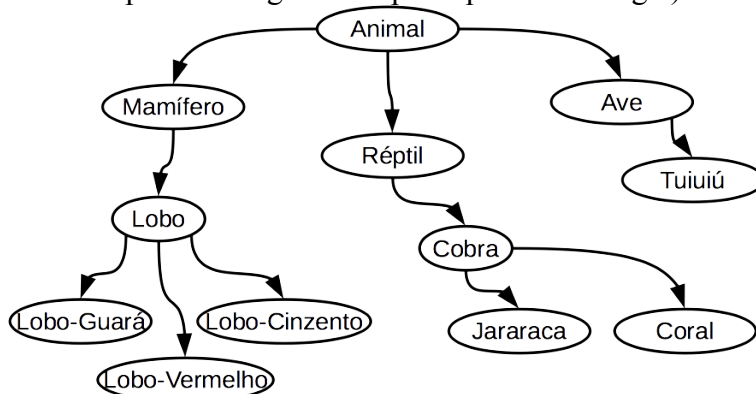
- (1) Contador de quantos registros/dados estão cadastrados na árvore;
- (2) Verificador da altura da árvore (retorna a altura da árvore);
- (3) Retornar um vetor contendo todos os elementos de um determinado nível da árvore (informado como parâmetro);
- (4) Imprimir a árvore por nível (todos os registros do 1º nível, seguido dos registros do 2º nível, e assim sucessivamente).

Problema 7 - Uma árvore de decisão é uma estrutura hierárquica que decide entre respostas (geralmente binária, "sim" ou "não") para chegar a um parecer/decisão. Considere um exemplo de árvore de decisão já elaborada, ilustrada na figura abaixo para decidir sobre a aprovação de um aluno:



Implemente um sistema que utilize a estrutura de dados *árvore* para representar a árvore de decisão apresentada acima e que então pergunte ao usuário sobre os itens de modo que à medida que for recebendo as respostas decida por qual ramo descer na árvore, até que se tenha um parecer/resposta a respeito da aprovação do aluno.

Problema 8 - O Zoológico de Barra do Garças está implementando um sistema para cadastro de seus animais. O cadastro deverá seguir a classificação biológica dos animais, segundo sua classe (mamífero, peixe, ave, réptil e anfíbio), família e raça. Considere a figura a baixo que ilustra um possível cadastro (tolere a imprecisão segundo os princípios da biologia):



Implemente o sistema que utilize a estrutura de dados *árvore* e que seja capaz de cadastrar tantos animais quanto for necessário. Os dados cadastrais devem ser informados em um arquivo de entrada onde cada linha do arquivo corresponde ao cadastro de um animal e deve ter a estrutura:

<classe>, <família>, <raça>

um exemplo de uma linha no arquivo, conforme figura acima, seria:

Mamíferos, Lobo, Lobo-Guará

Ao final implemente uma função que faça uma busca por um animal (informado pelo usuário) e então que imprima os dados do seu cadastro (conforme estrutura representada na árvore).

Problema 9 – Considere um jogo chamado **Direita e Esquerda**. Neste jogo, você joga primeiro e depois seu adversário. Você tem duas opções de movimento - esquerda ou direita. Posteriormente, seu concorrente também tem duas opções de reação - esquerda e direita. Dependendo das decisões, vocês vão ganhar uma quantia de "pontos". O objetivo de vocês é ganhar o máximo possível. O jogo tem apenas uma rodada.

- Se você escolher esquerda, e seu colega esquerda, você **ganha 12** e ele 8 pontos
- Se você escolher esquerda, e seu colega direita, você **ganha 5** e ele 10 pontos
- Se você escolher direita, e seu colega esquerda, você **ganha 8** e ele 20 pontos
- Se você escolher direita, e seu colega direita, você **ganha 10** e ele 2 pontos

Para melhor entender as opções, considere a figura abaixo:



Implemente o jogo de modo que a árvore de decisão acima seja representada em uma estrutura de dados em árvore. Você deve jogar contra o computador (adversário), que sempre joga após sua escolha inicial e que obviamente saberá da sua escolha e decidirá a jogada dele conforme a sua.

Lembre-se que esta é uma situação estratégica, onde a sua ação gera uma reação e o resultado depende da combinação das duas decisões - sua e do seu adversário. Então, qual a sua decisão (esquerda ou direita) que maximiza o seu resultado?