



Curso de Ciência da Computação
Disciplina: Estruturas de Dados
Professor: Josué Pereira de Castro

Trabalho para o 3ª Avaliação – Ano 2022

Instruções

1. Este trabalho vale 30 pontos para a 3ª Avaliação.
2. Este trabalho deve ser realizado por equipes de até dois (02) alunos.
3. Em nenhuma hipótese serão aceitas equipes com mais de dois alunos.
4. As equipes têm até 23h 30min do dia 31/07/2023 para submeter os trabalhos no sistema TEAMS. Após esta data e hora limite o sistema não aceitará mais submissões.
5. Não será aceita entrega de trabalho por outros meios além do especificado nestas instruções.
6. Caso ocorram casos de plágio, as equipes envolvidas receberão nota zero.
7. Este trabalho deve ser implementado em linguagem C++ ou Golang, em plataforma Linux/Unix.

Problema 1: Implementação de uma Agenda de Contatos indexada por Árvore-B (15 pts.)

Deverá ser implementado um aplicativo de agenda de contatos, que seja capaz de gerenciar os contatos armazenados em arquivo de dados indexado por uma árvore B, permitindo as seguintes operações:

1. Inclusão de novo contato: deve solicitar as informações sobre o novo contato e realizar a operação de inserção do novo contato no arquivo de dados, e atualizando também o arquivo de índice (que contém a árvore B)
2. Alteração dos dados de um contato: Deve permitir que os contatos tenham suas informações alteradas. Se ocorrerem alterações na chave de busca, estas devem se refletir também na árvore B e em seu respectivo arquivo de índice.
3. Listar contatos: Deve Produzir uma listagem dos dados armazenados de forma ordenada, sendo a ordenação realizada apenas pelo percurso em ordem na árvore B, recuperando-se os registros de contatos na ordem em que estes aparecem no percurso. O arquivo de dados não precisa ser ordenado.
4. Enviar para lixeira (Exclusão (Lógica): Esta operação deve marcar o contato selecionado para exclusão lógica, sem, no entanto, removê-lo fisicamente.
5. Retirar da Lixeira: Desmarca os contatos excluídos logicamente, "recuperando" suas informações.
6. Esvaziar Lixeira: Elimina fisicamente os registros de contatos marcados como apagados.

Descrição dos campos de dados a serem armazenados no arquivo de dados:

- Nome: Uma string com no máximo 30 caracteres (Chave de busca).
- Endereço: Uma string com no máximo 50 caracteres.
- Telefone: Uma string com no máximo 15 caracteres.
- Apagado: Um booleano que indica se o registro está ou não apagado (Na lixeira).

3. Descrição dos campos de dados no arquivo de índice

- chave: Nome
- posição (posição dentro do arquivo onde o registro correspondente se encontra).

4. Indexação:

Quando o programa inicia, o arquivo de índice deve ser lido para a memória e deve ser montada a árvore B. Qualquer operação deverá primeiro localizar o registro na árvore B, e então, de posse da

posição do registro desejado dentro do arquivo, deve-se realizar a operação desejada. Se for uma inclusão, esta será feita sempre ao final do arquivo de dados. Alterações serão feitas sempre na posição indicada pelo índice na árvore B. Remoções lógicas apenas marcam o registro como apagado. Remoções físicas implicam a reconstrução do índice inteiro. Alterações de chaves (nomes) implicam a reconstrução da árvore B em memória. Quando houver mudança nos índices, esta deverá ser refletida no arquivo de índices.

5. Itens Avaliados:

- | | |
|-------------------------------------|---------|
| a) Implementação da classe árvore B | |
| i) Inserção | (1 pt.) |
| ii) Remoção | (1 pt.) |
| iii) Busca | (1 pt.) |
| iv) Percurso em ordem | (1 pt.) |
| b) Funcionalidades da agenda | |
| i) Incluir contato | (1 pt.) |
| ii) Alterar contato | (2 pt.) |
| iii) Listar Contato | (2 pt.) |
| iv) Enviar para Lixeira | (2 pt.) |
| v) Recuperar da lixeira | (2 pt.) |
| vi) Esvaziar Lixeira | (2 pt.) |

Problema 2: Implementação de um sistema de manipulação e visualização de Grafos (15 pts.)

Deverá ser implementado um sistema de manipulação e visualização de grafos, orientado por menus, que realize as seguintes operações sobre grafos:

- Abrir/Salvar grafo: deve permitir abrir/salvar um arquivo texto com as informações referentes ao grafo e a partir destas informações montar o grafo em memória.
(1 pts.)
- Inserir vértice: Deve realizar a inserção de um novo vértice no grafo.
(1 pts.)
- Inserir Aresta: Deve realizar a inserção de uma aresta no grafo, entre vértices já existentes.
(1 pts.)
- Remover vértice: Deve realizar a remoção de um vértice, removendo também todas as arestas conectadas a ele.
(1 pts.)
- Remover aresta: Deve remover uma aresta do grafo, sem, no entanto, remover qualquer vértice ligado a mesma.
(1 pts.)
- Goodman (Componentes Conexos): deve calcular o número de componentes conexos do grafo e exibindo esta informação.
(1 pts.)
- Grafo Euleriano: Verificar se um grafo é Euleriano ou não, e exibir esta informação.
(1 pts.)
- Fleury (Ciclo Euleriano): Se o grafo for Euleriano, encontrar o ciclo euleriano utilizando o Algoritmo de Fleury, a partir de um nó informado.
(2 pts.)
- Busca em Profundidade: Deve realizar uma busca em profundidade no grafo e imprimir o trajeto encontrado (sequência de arestas percorridas) a partir de um nó informado.
(2 pts.)

- Busca em Largura: Deve realizar uma busca em largura no grafo e imprimir o trajeto encontrado (sequência de arestas percorridas) a partir de um nó informado.
(2 pts.)
- Dijkstra (Caminho de custo mínimo): Deve realizar o cálculo do caminho de menor custo entre um vértice de origem e um vértice de destino informado, exibindo a trajetória calculada.
(2 pts.)

Poderá ser utilizada qualquer forma de representação para os vértices e arestas.

Os vértices deverão ter as seguintes informações associadas:

- (a) Coordenadas (x, y): posição espacial do vértice na tela.
- (b) Rótulo: uma string que identifica o vértice.

As arestas deverão ter as seguintes informações associadas:

- (a) rótulo: uma string que identifica a aresta.
- (b) custo: representa o custo de se percorrer a aresta (p. ex.: distância em Km).

OBS: Após a realização de cada operação o sistema deverá imprimir a lista de arestas do grafo resultante atual.