Algoritmos e Estruturas de Dados II
Trabalho I: arquivos de dados com índices parciais

Objetivo: Criação de arquivos de dados e de arquivos de índice parcial para uma **organização sequencial-indexado**.

Organização: em duplas (ou individual) – se o trabalho for em dupla, definir seu grupo para a dupla no AVA.

Inicialmente cada equipe deverá criar o seu projeto no Git Hub para fazer as postagens de definição de contexto, perguntas (consultas), implementação de código, arquivos de dados e demais arquivos necessários.

Contexto do trabalho:

Arquivos de dados ou *datasets* são arquivos definidos e estruturados como **parte de uma organização de arquivos** que serão utilizados para consultas ou para alteração do conjunto de dados. Grandes volumes de dados são gerados a cada dia, e esses dados são de alguma forma guardados em arquivos, muitas vezes arquivos com grandes volumes de dados.

Conhecendo como um arquivo está organizado internamente, pode-se desenvolver programas ou procedimentos para consultar algum tipo de informação. Cada consulta é realizada para responder a uma pergunta específica:

Por exemplo, se a seguinte pergunta fosse relevante: *Qual é a joia mais vendida em um site de uma loja online de joias de médio porte?* Para responder a esta pergunta, pode-se criar uma base de dados a partir de extração dos dados de um site de *e-commerce*, ou pode-se utilizar um ou mais *datasets* entre os vários disponibilizados com acesso aberto na web (normalmente arquivos CSV), gerando um arquivo de dados único que contenha as informações que serão pesquisadas depois.

Nesse contexto, a partir da pergunta formulada, seria possível estabelecer algumas **hipóteses** (cada hipótese é o que eu acho que poderia acontecer, possíveis respostas para minhas perguntas):

- é possível que um tipo de joia seja líder absoluto de vendas, com mais de 50% das compras;
- é possível que haja um equilíbrio de vendas entre algumas joias;
- etc.

A partir deste contexto, o próximo passo é extrair as informações e montar uma base de dados: o seu arquivo de dados. Para definir a estrutura da base de dados, é necessário definir quais as informações serão relevantes incluir nessa base, e o que se tem de dados disponíveis na ou nas fontes disponíveis.

Atividades a realizar

1. **Definição do contexto a ser explorado:**

O *dataset* contém dados de compras de dezembro de 2018 a dezembro de 2021 (3 anos) em uma loja *online* de joias de médio porte. Cada linha do arquivo representa um produto comprado. Vários produtos do mesmo pedido/compra são listados em linhas separadas e unidos pelo campo order_id.

O dataset para trabalho de organização de arquivos está em:
https://www.kaggle.com/datasets/mkechinov/ecommerce-purchase-history-from-jewelry-store/data

2. **Montagem dos arquivo de dados**

A primeira atividade do trabalho envolve a construção **dos arquivos de dados**. O *dataset* a ser utilizado tem dados d**e compras em uma loja online** realizadas durante 3 anos, e contém: data e hora do pedido, id do pedido, id do produto adquirido, quantidade de SKU[1] no pedido, id de categoria, alias de categoria, identificação da marca, preço em U$D, id do usuário, gênero do produto. **Você deve criar no mínimo dois arquivos com esses dados: arquivo de joias (cadastro), e arquivo de acesso compras (pedidos).**

Como a **organização de arquivos de dados** definida **é sequencial**, os arquivos devem estar ordenados por algum dos campos, preferencialmente o campo com um identificador (campo **chave**). Assim, as seguintes tarefas deverão ser realizadas:

- Escolher os arquivos que serão criados com o *dataset* fornecido;
- Cada arquivo deve ter pelo menos 3 campos (colunas) de informações: pelo menos um dos campos com dados não repetidos (o campo da **chave**), e pelo menos um dos campos com informações repetidas;
- Definir duas ou três perguntas que se poderia fazer a esse conjunto de dados (serão as consultas que serão realizadas nos dados) – **definir consultas simples, lembre-se que é um arquivo e não um banco de dados**;
- Ordenar os dados do arquivo de dados pelo campo **chave** (que não tem dados repetidos). Pense em utilizar algum dos métodos de ordenação trabalhados em aula;
- Os arquivos de dados devem ser criados em modo binário (não textual).

2.1) Organização e registros do Arquivo de Dados:

1Stock Keeping Unit, ou Unidade de Manutenção de Estoque, e é um código alfanumérico interno que uma empresa cria para identificar de forma única cada variação de um produto em seu estoque, como cor, tamanho ou modelo.

Os registros dos arquivos de dados devem ser de **tamanho fixo**. Para a implementação dessa funcionalidade, deve-se inserir espaços em branco no final dos dados textuais se necessário, para que os textos fiquem todos do mesmo tamanho.

Cada linha do arquivo é encerrada com com o caractere '\n'. A implementação deve ser feita em uma linguagem de programação (C, C#, C++, Python, PHP, Java ...) que possua o comando *seek* ou similar.

- Implementar, para cada arquivo de dados:
    1. uma função para inserir os dados: explicar como os dados foram ordenados (se for o caso) e inseridos;
    2. **uma função para mostrar os dados**,
    3. **uma função para realizar a pesquisa binária** e
    4. **uma função para consultar dados a partir da pesquisa binária**.

Deverão ser construídos 2 **índices parciais**, um para cada arquivo de dados (salvos em arquivo no final da execução de um programa, e carregados quando o programa for aberto).

2.2) Índices em arquivo:

- Implemente **um arquivo de índice parcial** para o campo **chave** de cada arquivo de dados de acordo com a descrição do índice de arquivo da organização sequencial-indexado;
- **Implemente uma função de consulta a partir deste índice** usando a **pesquisa binária** para pesquisar no arquivo de índice e, depois o comando *seek* para pesquisar no arquivo de dados.

3. Inserção/remoção de dados em um dos arquivos de dados, e reconstrução do índice:

- Como será gerenciada a inserção de um novo registro no arquivo de dados?
- Como será gerenciada a remoção de um registro no arquivo de dados?

Implemente operações de inserção e remoção de registros em um dos arquivos, o que vai acarretar reconstrução do índice daquele arquivo. Definir se a reconstrução do índice ocorre a cada inserção/remoção, ou se seguirá algum outro critério.

4. Postar no AVA:

- Descrição dos arquivos de dados e descrição dos arquivos de índices.

- Link para o projeto no GiT Hub, onde deve estar: o código-fonte da implementação, os arquivos de dados, os arquivos de índices gerados para aqueles dados.

Avaliação:

- O trabalho vale 10 pontos e será avaliado conforme o cumprimento das atividades propostas e a utilização de boas práticas de programação.
- Não é permitido o uso da memória RAM para armazenar todos (ou grande parte) dos registros **do arquivo de dados** para efetuar as buscas, devem ser trazidos para a memória apenas os dados necessários. Todas as operações solicitadas devem ser executadas no arquivo de dados armazenado em memória secundária.