Algoritmos e Estruturas de Dados II
Trabalho I

Objetivo: Criação de um arquivo de dados de **organização sequencial-indexado**, para o qual serão construídos **4 índices**: dois índices de arquivo, e dois índices de memória.

Especificação do Trabalho I:

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.

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

Explicação geral:

Arquivos de dados ou *datasets* são arquivos definidos e estruturados como parte de uma organização de arquivos de forma a poderem ser 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 medida preventiva para a COVID-19 que é mais comentada no Brasil na rede social Twitter?* Para responder a esta pergunta, pode-se criar uma base de dados a partir de extração dos *tweets* pesquisando pelo usuário correspondente @COVID19 ou por palavras chaves, tais como, #covid19prevencao, #covid_19brasil, bem como, pesquisar nas páginas do Ministério da Saúde @minsaude , Secretaria da Saúde do RS @SES_RS, entre outras. 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 a medida preventiva mais comentada seja o uso de máscaras
- é possível que a medida preventiva mais comentada seja o distanciamento social
- 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. Para o contexto apresentado como exemplo

(medidas preventivas para COVID-19) poderia se considerar relevantes *o nome do usuário que postou a mensagem*, *a mensagem em si*, *o local de postagem*, *as palavras de busca, a data, etc*.

Após a base de dados ser construída, ainda considerando o exemplo apresentado, pode-se fazer uma pesquisa pelo tipo de prevenção, se é de atitudes, de uso de máscara, de distanciamento social, etc.

Atividades a realizar

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

O contexto dos dados é de sua livre escolha. Procure dados em *datasets* disponíveis na web sobre o contexto que você deseja explorar, ou extraia os dados de redes sociais ou de bancos de dados. Se não tiver dados suficientes, gere os dados necessários para completar sua base.

2) **Montagem do arquivo de dados**

A primeira atividade do trabalho envolve a construção do arquivo de dados. Como a organização de arquivos definida é sequencial, o arquivo deve estar ordenado por algum dos campos, preferencialmente o campo com um identificador (campo **chave**). Assim, as seguintes tarefas deverão ser realizadas:

- Escolha um ou mais conjuntos de dados, para obter um arquivo textual de pelo menos 100 000 registros. É possível utilizar arquivos já prontos e formatados, ou extrair registros de algum lugar (de uma rede social por exemplo) e tratá-los para ficarem no formato adequado;
- Cada registro desse arquivo textual deve ter pelo menos 4 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. Esses campos serão referenciados nesse texto como **campo 1**, **campo 2**, **campo 3** e **campo 4**. Exemplo de campos: código, nome/título, localização/cidade, data, idade, pontos, valor, etc;
- Definir uma ou mais perguntas (serão as consultas que serão realizadas nos dados). Para responder essa(s) pergunta(s), poderá ser utilizado um dos índices construídos ou não;
- Ordenar os dados do arquivo de dados pelo campo **chave** (que não tem dados repetidos). É possível gerar as chaves pelo incremento de um número sequencial.

2.1) Organização e registros do Arquivo de Dados:

Os registros do arquivo 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:
    a. uma função para inserir as 100 mil (+-) linhas de dados: explicar como os dados foram ordenados (se for o caso) e inseridos;
    b. **uma função para mostrar os dados**,
    c. **uma função para realizar a pesquisa binária** e
    d. **uma função para consultar dados a partir da pesquisa binária**.

Deverão ser construídos **4 índices**, dois em arquivos (salvos em arquivo no final da execução de um programa, e carregados quando o programa for aberto) e dois em memória (a serem construídos em memória cada vez que o for iniciada a execução do programa). **Cada índice será construído sobre um campo diferente do arquivo de dados, e o programa deverá ter opções de consulta aos dados com cada um dos tipos de índice**.

2.2) Índices em arquivo:

- Implemente **um arquivo de índice** para o campo **chave** (campo 1) 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.

- Implemente **um arquivo de índice** para um outro campo **que não seja o campo chave** (campo 2) 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.

2.3) Índices em memória:

- Implemente uma estrutura de índice em memória *(a organização do índice é a sua escolha)* para um campo não utilizado para os índices de arquivo (campo 3) e **que tenha valores repetidos**. Implemente um procedimento de consulta a partir deste índice e, depois o comando *seek* para pesquisar no arquivo de dados.

- Implemente uma estrutura de **árvore de pesquisa (binária ou alguma árvore balanceada)** para um outro campo não utilizado para outros índices (campo 4). Implemente um procedimento de consulta a partir deste índice e, depois o comando *seek* para pesquisar no arquivo de dados.

3) Resposta à(s) pergunta(s), para confirmar ou não as hipóteses levantadas:

Implemente um procedimento/função para responder a pergunta (ou as perguntas) definida no início do trabalho (dependendo da pergunta, pode ser necessário usar ou não seus índices).

4) **Postar no AVA**:

- Código fonte (ou códigos-fonte), descrição do contexto escolhido, a descrição da(s) hipótese(s), descrição da rede social da qual foram extraídos os dados, período de extração dos dados.

- Link para o projeto no GiT Hub, onde deve estar: a descrição do contexto escolhido, a descrição da(s) pergunta(s), o arquivo 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 (disco rígido e similares).