CIC4002 Algoritmos e Estruturas de Dados II
Trabalho I - 2023/4

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:

Atividade em duplas (ou individual).

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

Explicação geral:

Arquivos de dados são arquivos definidos e estruturados como parte de uma organização de arquivos, para serem 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 muitos 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. Cada consulta pode utilizar um ou mais índices para tornar a consulta mais rápida.

Atividades a realizar

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

O contexto dos dados é **Aplicativos da *PlayStore***, e o *dataset* para o trabalho está em:

https://www.kaggle.com/datasets/gauthamp10/google-playstore-apps

676.46Mb, 1 arquivo csv
23 colunas:
2312944 linhas

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**). Modifique o *dataset* para a construção do arquivo de dados como for necessário, para realizar as seguintes tarefas::

- Verificar o campo que define a chave (se está ok, se não há registros duplicados) no *dataset* (arquivo CSV). É preciso que exista um campo chave sem registros duplicados;

- Analisar o conjunto de dados (ou ao menos dar uma olhada nos valores dos dados das colunas). Cada registro (linha de dados) desse arquivo textual deve ter pelo menos 4 campos (colunas) de informações que possam ser usadas como índices: 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: identificação, nome/título, localização/cidade, categoria, classificação, número de instalações, valor, etc;

- Definir uma ou mais perguntas (serão as consultas que serão realizadas nos dados). Para responder essa(s) pergunta(s), serão utilizados os índices construídos;

- Ordenar (ou verificar se estão ordenados) os dados do arquivo de dados pelo campo **chave** (que não tem dados repetidos). A ordenação já pode ser realizada no arquivo textual (csv), e pode ser feita com alguma ferramenta (como uma planilha);

- Inserir os registros ordenados em um arquivo binário. 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, antes de salvar no arquivo binário. 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 o arquivo de dados):
    a. uma função para inserir as linhas de dados no arquivo binário (vai ser utilizada uma vez apenas): explicar como os dados foram ordenados (se for o caso) e inseridos;
    b. **uma função para realizar a pesquisa binária** e
    c. **uma função para consultar dados a partir da pesquisa binária**.

3) Montagem dos índices

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**.

3.1) 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: esse índice deve ser parcial, uma vez que o arquivo de dados já está ordenado. **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: esse índice deve ser sobre um campo não ordenado no arquivo de dados (a ordenação vai ser do índice). **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.

3.2) 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 (AVL 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.

4) Postar no AVA:

- **1)** Código fonte (ou códigos-fonte), **2)** a descrição das operações implementadas e dos índices construídos (sobre quais campos cada um foi construído, e qual estrutura foi utilizada para cada um dos 4 índices), **3)** descrição das perguntas/consultas que podem ser realizadas com os índices.

- **Link para o projeto no GiT Hub**, onde deve estar: 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, as estruturas escolhidas e implementações realizadas para os índices, 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).