



2º Trabalho Prático

Assunto: Árvores B

Informações Gerais

A principal aplicação de árvores B é a indexação. Servidores de banco de dados (SGBD) utilizam essa estrutura como padrão para criar os índices das tabelas. No caso do banco de dados, índice é um arquivo ordenado associado ao arquivo que guarda os dados das tabelas (arquivos de dados). Quando o usuário faz uma consulta ao banco de dados, o SGBD consulta o arquivo de índice para encontrar mais facilmente o dado pesquisado. Nesse caso, o índice é um arquivo composto por pares de chave e ponteiro. Esse ponteiro guarda o endereço do dado no disco. O índice só ajuda se a consulta for realizada com a chave indexada.

A Figura 1 ilustra o funcionamento de um índice. No caso ad table é a tabela de dados, armazenada no disco rígido. O arquivo index é o arquivo de índice que guarda as chaves, que representam valores do atributo company_num, e o endereço dessa chave no disco. O arquivo de índice é sempre ordenado. A Figura 1 ilustra um índice mononível, que ajuda no entendimento da função do índice. No entanto, na prática, os índices são multiníveis e, por isso, implementados com B-Tree.

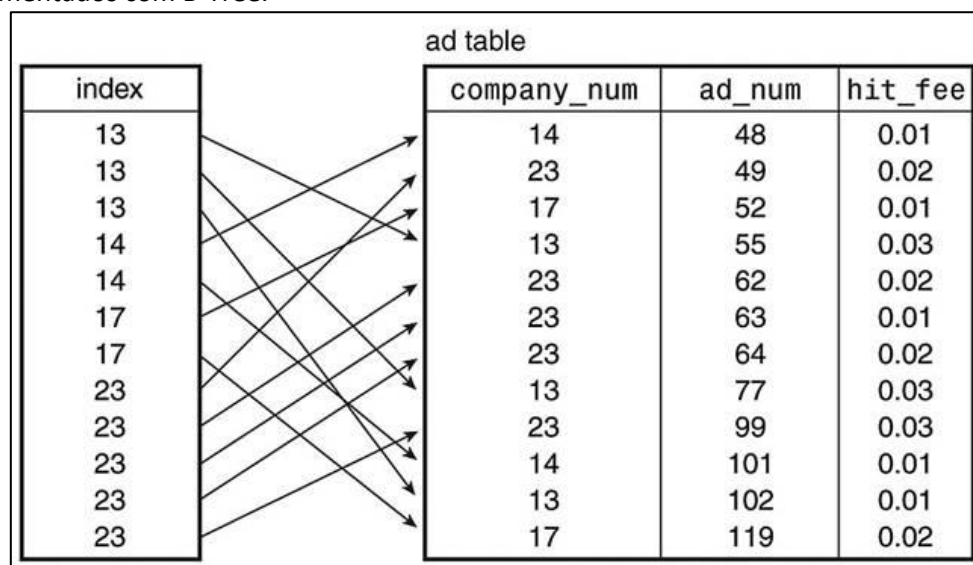


Figura 1: Ilustração do funcionamento de um índice em um banco de dados.

O objetivo desse trabalho é simular o índice de banco de dados em memória RAM. Para tanto, a implementação da B-Tree será feita em memória RAM e o endereço do registro será a linha do arquivo lido.



Desenvolvimento

A implementação deverá ser na linguagem de programação C e propiciar as seguintes operações:

- Recebe o arquivo de entrada que simula uma tabela no banco de dados. O arquivo deve conter registros com, pelo menos, quatro atributos. O atributo que será utilizado como índice deve ser numérico e não possuir valores repetidos (representa uma chave primária).
- O programa deve ler o arquivo e criar uma B-Tree para a chave primária. Esse fluxo está identificado na Figura 2.
- O usuário poderá procurar elementos nesse arquivo a partir da chave primária. Nesse caso, o programa deverá procurar o elemento na B-Tree e retornar o valor associado à chave. Esse fluxo está identificado na Figura 3.
- O usuário poderá remover um elemento do índice. Considere a remoção apenas no índice (B-Tree). Ou seja, o dado se mantém no arquivo de entrada.

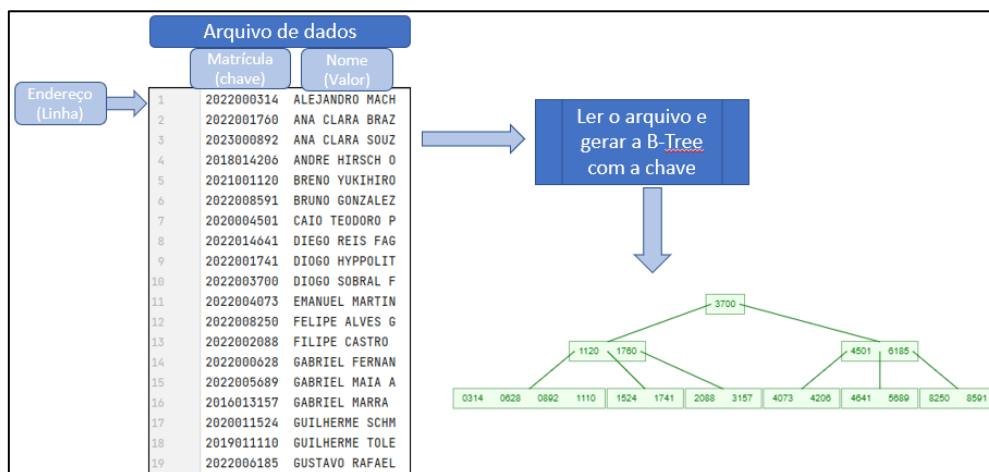


Figura 2 : Geração da B-Tree a partir da chave primária do arquivo. ATENÇÃO: Árvore gerada com os 4 últimos dígitos da matrícula. Apenas ilustrativo.

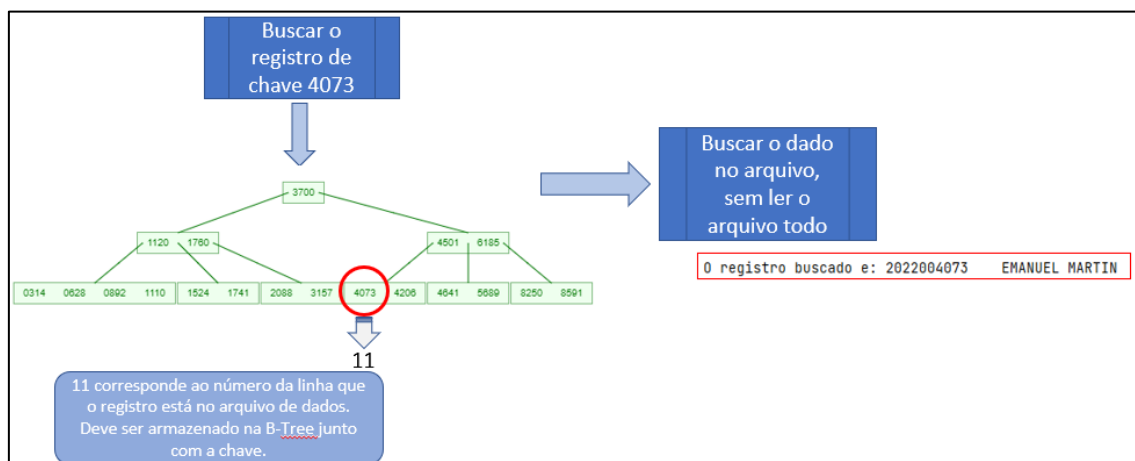


Figura 3: Pesquisa na B-Tree e retorno do registro. O registro deve ser retornado com base na linha do arquivo, sem ter que procurar pelo arquivo todo.



O programa principal deve ser organizado da seguinte maneira:

Menu

1. Criar índice
2. Procurar elementos
3. Remover registro
4. Sair

Análise

O grupo deverá analisar se a busca utilizando o índice da B-Tree performa melhor do que a busca direto no arquivo. Para tanto, deverá realizar 30 buscas aleatórias e obter a média, valor máximo e valor mínimo do tempo gasto. Para esse teste, o grupo deverá gerar um arquivo com dados aleatórios com mais de 10 mil registros. Mostrar o gráfico comparando o tempo de consultar direto no arquivo e utilizando a B-Tree.

Atenção: para esse teste, não considerar o tempo de geração da B-Tree. Apenas o tempo de consulta.

Entrega

Os grupos deverão postar os códigos, relatórios e apresentação até o dia **03/07/2023, às 13h**. Os grupos apresentarão o trabalho para a professora, por sorteio, nos dias **03 e 05/07**.

O relatório e apresentação devem conter:

- Estratégia utilizada na implementação da B-Tree (inserção e remoção)
- Estratégia utilizada para leitura direta no arquivo de dados pós busca na B-Tree
- Resultados da análise
- Referências utilizadas

Na apresentação, incluir:

- Principais dificuldades do grupo
- Principais aprendizados do grupo

Distribuição de Pontos

A distribuição dos pontos é como se segue:

- O fluxo Ler Arquivo -> Gerar B-Tree valerá 25% da nota
- Pesquisa na B-Tree -> 25%
- Remoção na B-Tree -> 20%
- Relatório -> 15%
- Apresentação -> 15%



Considerações Importantes

- Para conseguir ler a linha do arquivo diretamente, é importante que todos os registros tenham o mesmo tamanho.
- A ordem da árvore deve ser informada pelo usuário, mas o grupo pode definir um valor máximo.
- Trabalhos plagiados, tanto da internet (incluindo chatGPT), quanto de colegas, serão zerados. Nesse trabalho, o grupo deverá implementar seus próprios códigos.
- Na apresentação do trabalho os integrantes do grupo serão arguidos individualmente sobre detalhes do programa. Caso algum integrante não saiba responder, o mesmo perderá a nota total do trabalho.
- Colocar no relatório todas as referências utilizadas para o desenvolvimento do trabalho.
- As apresentações serão só para a professora, em horários definidos. Grupos que não chegarem no horário terão desconto de nota.