

Índices Invertidos

Orientações:

Nesta prática em laboratório, você deverá codificar e testar a implementação de um índice invertido de um arquivo de dados utilizando uma tabela hash em disco como índice. Você deve usar encadeamento exterior para resolver as eventuais colisões (veja código no final do enunciado).

Você deverá criar um arquivo de dados (use a biblioteca Random Access File do Java) para armazenar cada registro do arquivo da base. As consultas serão realizadas pelo campo por um campo do tipo "string". Ou seja, a consulta será realizada em um campo texto e deverá ser permitido consulta por qualquer palavra que possa conter dentro do texto deste campo. Para permitir a consulta neste registro, você deverá criar um arquivo invertido com todas as palavras contidas na lista de palavras presentes neste campo (de todos os registros). Atente-se para eliminar palavras muito comuns. Você também deve criar um índice utilizando hash para o seu arquivo invertido (este índice deve estar armazenado em um arquivo). Conforme estudando em sala, a consulta por palavra(s) deverá ser realizada no índice e não diretamente no arquivo invertido. Para representar a palavra em forma de número para aplicar a função hash, você pode utilizar qualquer estratégia (veja exemplo nas notas de aula). Importante: os arquivos utilizados (arquivo de dados, arquivo de lista invertida e arquivo de índice) não devem ser carregados em memória para a realização da consulta.

Você deverá codificar os seguintes métodos:

1. Inicialização (construtora) : responsável por estabelecer um estado inicial "bem definido" para a estrutura de dados;
2. insere(novoElemento, chave) : responsável por armazenar um novo elemento juntamente com o valor de sua chave (que será utilizado mais tarde para localizar o elemento);
3. pesquisa(chave) : responsável por localizar no conjunto o elemento cuja chave for passada como parâmetro e retorná-lo para o "cliente" do TAD;
4. imprime() : responsável por imprimir os elementos da tabela hash, bem como suas chaves, na ordem em que se encontram armazenados.

Além disso, você deve codificar um programa para teste das implementações realizadas. Seu programa deve criar e inserir e pesquisar (não precisa implementar a remoção) uma mesma série de elementos lidos do arquivo (arquivo .csv que será fornecido pelo professor). Por fim, modifique suas implementações de modo a contabilizar o número de comparações realizadas durante as operações de pesquisa.

O trabalho deverá ser entregue no SGA e **apresentado em sala para o professor. Trabalhos que não forem apresentados não receberão nota.**