UNIFOR
UNIVERSIDADE DE FORTALEZA
FUNDAÇÃO EDSON QUEIROZ
ENSINANDO E APRENDENDO

ASSINALE A VERIFICAÇÃO CORRESPONDENTE A ESSE TRABALHO: AV 1 ☐ AV 2 ☐ AV 3 ☐

CENTRO:_____ DISCIPLINA:_____ DATA: ___/___/___ PROF(ª):_____

ALUNO(A): _____ MATRÍCULA: _____

ALUNO(A): _____ MATRÍCULA: _____

ALUNO(A): _____ MATRÍCULA: _____

INSTRUÇÕES PARA O DESENVOLVIMENTO DO TRABALHO

- Trabalho é em Equipe (até 4 pessoas)
- O Trabalho vale 30% da nota da disciplina.
- O Trabalho será apresentado pela equipe. Será marcado um dia para a equipe mostrar a implementação e a execução da aplicação.

Implementar um índice hash estático

1. Interface gráfica obrigatória ilustrando as estruturas de dados e o funcionamento de um índice hash estático.

2. Funcionalidades principais:

a. Construção do índice;
b. Busca por uma tupla a partir da entrada de uma chave de busca usando o índice construído;

3**. Entidades/estruturas a serem implementadas (usando POO como padrão):**

a. **Tupla:** representa uma linha da tabela, contém o valor da chave de busca e os dados da linha.
b. **Tabela:** contém todas as tuplas construídas a partir do carregamento do arquivo de dados.
c. **Página**: estrutura de dados que representa a divisão e alocação física da tabela na mídia de armazenamento.
d. **Bucket**: estrutura de dados que mapeia chaves de busca em endereços de páginas.
e. **Função hash**: mapeia uma chave de busca em um endereço de bucket. Deve ser escolhida/projetada pela equipe.

4. Parâmetros:

a. **Arquivos de dados:** contém os dados que serão usados para popular uma ou mais tabelas. Para este trabalho será usado um arquivo com 479 mil palavras do idioma Inglês, disponível em: https://github.com/dwyl/english-words
b. **Tamanho da página:** entrada de usuário que determina o tamanho de cada página individualmente.
c. **Quantidade de páginas**: número máximo de páginas usadas [1] para dividir o conteúdo da tabela.
d. **Número de buckets (NB):** calculado, onde NB > NR / FR. NR é a cardinalidade da tabela (número de tuplas) e FR é o número de tuplas endereçadas por bucket.
e. **Tamanho dos buckets (FR):** número máximo de tuplas endereçadas por bucket, depende da função hash implementada.

[1] **Se o usuário escolheu o tamanho da página então a quantidade de páginas é um parâmetro calculado. Os dois parâmetros são mutuamente exclusivos como entrada.**

f. **Chave de busca de uma tupla específica:** depois que o índice é construído o usuário pode entrar com uma chave de busca para que o sistema retorne a tupla associada.

5. Problemas:

a. A implementação do índice deve levar em consideração as colisões, ou seja, deve ser implementado um algoritmo de resolução de colisões.

b. A implementação do índice deve levar em consideração o transbordamento dos buckets (bucket overflow), ou seja, deve ser implementado um algoritmo de resolução de overflow.

6. Estatísticas:

a. Deve ser calculada e mostrada a taxa de colisões.

b. Deve ser calculada e mostrada a taxa de overflows.

c. Deve ser calculado e mostrado uma estimativa de custo (acessos a disco), quando uma chave de busca é entrada (funcionalidade b).

7. Funcionamento em passos:

a. O arquivo de dados é carregado em memória.

b. Cada linha do arquivo deve gerar uma tupla, que será adicionada à tabela.

c. As tuplas da tabela devem ser divididas em páginas, de acordo com o tamanho das páginas.

d. NB buckets de tamanho FR são criados.

e. A função hash é aplicada à chave de busca de cada tupla; a chave de busca e o endereço da página onde a tupla foi armazenada são adicionadas ao bucket cujo endereço foi calculado pela função hash.