Universidade Federal do Maranhão
Departamento de Informática
Disciplina: ***Estrutura de Dados II 2024.1***
Prof.: ***João Dallyson Sousa de Almeida***

Atividade Prática 2 (DUPLA)
Valor: 40% da 2ª Avaliação

Problema 1:

Nesta atividade iremos implementar Índices Remissivos para arquivos texto de entrada. Seu programa deverá inserir as palavras com mais de X caracteres no índice remissivo armazenando a informação das linhas em que as palavras ocorrem no texto. X é um parâmetro fornecido pelo usuário. Cada palavra pesquisada (busca) nos índices deverá retornar as linhas de cada ocorrência e quanto tempo levou para retornar a informação do termo pesquisado. O tempo corresponde a quantidade de passos necessários para encontrar o termo.

a) O seu programa deverá armazenar em um índice remissivo implementado utilizando Hash, Árvore Rubro-Negra e Árvore AVL.
b) A tabela Hash deverá ser implementada conforme visto em sala de aula, utilizando um flag para controle da ocupação do espaço.
c) O Hash deverá ter um tamanho inicial fixo e deverá ser dobrado sempre após estiver com um %X de espaços ocupados. Isso implicará na criação de um novo Dicionário, cópia dos itens para o novo dicionário e remoção do dicionário anterior.
d) Seu programa deverá permitir a escolha das seguintes estratégias para tratamento de colisões: tentativa linear ou tentativa quadrática.
e) Você deverá implementar uma versão adaptada da árvore AVL que permitirá controlar o limite do Fator de Balanceamento para a realização das operações de rotação. Neste caso, a árvore permitirá "D" diferença de altura entre subárvores.
f) Seu programa deverá permitir buscar palavras do índice: Dada uma palavra e o Índice (Hash, AVL ou RB), seu programa deverá retornar ao usuário a informação das linhas onde em palavra ocorre no texto.
g) Seu programa deverá permitir a remoção de palavras do índice.
h) Imprimir índice: O usuário escolherá qual índice irá imprimir. Após a escolha, seu programa deverá imprimir o índice em ordenado por termo com suas respectivas ocorrências.

Problema 2:

Implemente uma árvore B funcional que suporte operações de inserção, exclusão e busca. O projeto deve incluir a capacidade de lidar com dados em memória secundária simulada (por exemplo, utilizando arquivos de texto em disco.

a) Definir a estrutura dos nós da árvore, incluindo referência para filhos conforme o parâmetro t estudo em aula.
b) Implemente as operações básicas da árvore B, como inserção, exclusão e busca.
c) Certifique-se de considerar como lidar com split e merge de nós durante a inserção e a exclusão para manter as propriedades da árvore B.
d) Simule o armazenamento dos dados em memória secundária (disco) usando arquivos.
e) Desenvolva funções para salvar e carregar nós da árvore em arquivos, representando a hierarquia da árvore B.
f) Utilize a Árvore B para armazenar dados (inserir, buscar e excluir) de uma tabela de uma banco de dados. Por exemplo, uma tabela Usuario. A árvore B usará a chave primária (id_usuario). Os campos da tabela usuário são: id_usuario, login, nome, email, data_nascimento e foto.

g) Permitir que o usuário insira novos dados, busque informações na árvore e visualize a estrutura dos nós.

Experimentos:

1) Todos o métodos deverão implementar os objetos utilizando *Generics (https://docs.oracle.com/javase/tutorial/java/generics/types.html).*

2) Você deverá enviar, também, um relatório mostrando e discutindo os resultados obtidos. Mostre prints da execução e dos resultados.

3. Entrega

- Código fonte do programa em JAVA (bem identado e comentado) utilizando os conceitos de Orientação a Objetos. Relatório com os resultados (Upload no SIGAA)