



UFAM

**Universidade Federal do Amazonas
Instituto de Computação**

**Algoritmos e Estruturas de Dados 2 (AED2)
Trabalho Prático 3 - entrega até 05 e 07/06/2019**

Este trabalho trata de estruturas de dados para indexação de um arquivo em disco. Deverão ser implementados três índices: dois usando árvore binária de pesquisa e outro usando uma tabela *hash*. Também deve ser implementada a busca sequencial diretamente no arquivo.

Importante: não esqueçam de desenvolver os programas usando os conceitos de tipos abstratos de dados, conforme foi feito nos trabalhos práticos anteriores.

O arquivo deve possuir um campo chave (valores únicos, não-repetidos) e mais 4 ou 5 atributos. Um dos atributos não-chave deve ser usado para consultas do tipo “maior que” e “menor que”. Como exemplo, o arquivo pode ser um cadastro de alunos, produtos de uma loja, clientes de uma empresa, etc. O arquivo deve ser criado como uma sequência de registros, com tamanho suficiente para que o tempo de uma operação de busca possa ser medido. Faça testes com uma quantidade menor de elementos (ex: 100) e aumente a quantidade para que o teste de busca retorne tempos de execução mensuráveis.

Devem ser criados dois índices para o atributo chave: um usando árvore binária de pesquisa e o outro usando tabela *hash*. O objetivo é comparar os tempos de consulta usando essas duas estruturas de dados. Adicionalmente implemente a busca sequencial no disco pelo atributo chave. Ao criar a tabela *hash*, mostre a quantidade de colisões.

O terceiro índice será usado para indexar o atributo que será usado para as consultas do tipo “maior que” ou “menor que”. Para essa tarefa deve ser implementada uma árvore binária de pesquisa. O resultado será uma lista (sequência) de registros, mostrando todos os dados de cada item que atenda às condições da consulta. Adicionalmente implemente a busca sequencial pelo mesmo atributo.

Resumindo as estratégias de busca a serem implementadas:

- 1) Busca usando a árvore binária de pesquisa para o atributo chave;
- 2) Busca usando *hashing* para o atributo chave;
- 3) Busca sequencial no arquivo para o atributo chave;
- 4) Busca do tipo $>$, \geq , $<$ ou \leq usando a árvore binária de pesquisa para um atributo não-chave.
- 5) Busca do tipo $>$, \geq , $<$ ou \leq usando busca sequencial no arquivo para o mesmo atributo do item 4.

Para comparar o tempo das cinco formas de busca para o atributo chave e para o atributo das consultas do tipo “maior” ou “menor”, execute cada busca pelo menos 30 vezes usando cada uma das cinco estratégias de busca.

Utilize 30 valores distintos e execute a consulta usando o mesmo valor em cada uma das três estratégias de busca pelo atributo chave.

Escolha outros 30 valores distintos para as consultas do tipo “maior” ou “menor” e execute as 30 consultas usando os mesmos valores para a busca usando o índice e para a busca sequencial no arquivo.

Calcule o tempo de execução para cada busca e a média entre as 30 execuções para cada estratégia de busca.