



Implementação de Tabelas Hash com Diferentes Funções Hash

Objetivo:

O objetivo desta atividade é implementar duas tabelas hash, utilizando duas funções hash diferentes, e compará-las quanto à eficiência em termos de:

- Número de colisões.
- Tempo de inserção e busca.
- Distribuição das chaves na tabela.

Descrição:

Você deve criar um programa em Java que utilize **Programação Orientada a Objetos (POO)** para modelar as tabelas hash. As tabelas devem seguir o seguinte fluxo:

1. **Leitura de um arquivo CSV** contendo uma lista de 5000 nomes (arquivo disponibilizado).
2. **Inserção dos dados** (os nomes) nas duas tabelas hash.
3. Realização de testes de eficiência nas tabelas (detalhes a seguir).
4. Impressão de relatórios comparativos no console.

Detalhes do Trabalho:

- **Funções Hash:** Implemente duas funções hash distintas. Você pode usar ou adaptar funções hash prontas ou criar a sua própria.
- **Tratamento de Colisões:** Escolha um método de tratamento de colisões discutido em sala.
- **Orientação a Objetos:** Utilize uma **classe abstrata** para representar a tabela hash de forma genérica, e duas classes específicas que implementam a tabela com as diferentes funções hash. Cada classe deve sobrescrever apenas a função hash utilizada.

Fluxo do Programa:

1. **Inicialização:** O programa deve iniciar lendo um arquivo CSV contendo 5000 nomes (arquivo names_5000.csv).
2. **Inserção:** Insira os nomes na tabela hash utilizando as duas funções hash implementadas.



Pontifícia Universidade Católica do Paraná

Resolução de Problemas Estruturados em Computação

Bacharelado em Engenharia de Software

Avaliação Somativa – TDE 03

Prof.^a Marina de Lara

- A inserção pode ser feita em uma tabela de cada vez ou em ambas simultaneamente, a critério do aluno.

3. Testes de Eficiência:

- **Número de colisões:** Contabilize quantas colisões ocorreram ao longo das inserções.
- **Tempo de inserção e busca:** Meça o tempo para inserir todas as chaves e realizar buscas em um conjunto de nomes.
- **Distribuição das chaves:** Verifique a distribuição das chaves nas tabelas (quantidade de chaves em cada posição).

4. Relatório Final: O programa deve gerar e imprimir no console um relatório contendo:

- O número de colisões para cada tabela.
- O tempo total de inserção e busca para cada função hash.
- A distribuição das chaves (quantidade de chaves em cada posição da tabela).

O que deve ser entregue:

- Código-fonte Java organizado conforme POO.
- Relatório em PDF com os dados gerados no console, incluindo:
 - Número de colisões para cada função hash.
 - Número de colisões para cada posição (clusterização) com cada função hash.
 - Tempos de inserção e busca