Universidade Federal de Lavras

ICTIN - Instituto de Ciência, Tecnologia e Inovação

Autor:

Enzo Augusto Valentim

Sistema de Gerenciamento de Estoque TechParts

São Sebastião do Paraíso

Setembro/2025

Introdução:

O presente trabalho desenvolvido na disciplina de Estruturas de Dados II e teve o objetivo de aplicar os conceitos estudados sobre tabela hash em um problema prático de gerenciamento de estoque. A empresa fictícia TechParts, utilizada como cenário, enfrenta dificuldades em organizar e consultar mais de 10 mil peças cadastradas, o que torna os processos lentos e propensos a erros.

Para resolver esse desafio, foi implementado em linguagem C um sistema de gerenciamento de estoque baseado em tabelas hash com encadeamento separado, garantindo eficiência nas operações de inserção, busca e remoção. Além disso, o sistema conta com recursos adicionais, como salvamento e carregamento de dados em arquivos CSV, rehash automático e a possibilidade de alternar entre diferentes funções de dispersão para comparar o desempenho.

Modelagem do Problema e métodos utilizados:

Para representar o estoque da empresa TechParts, foi utilizada a estrutura de tabela hash com encadeamento separado. Cada bucket da tabela corresponde a uma lista ligada, onde são armazenadas as peças cujos códigos foram mapeados para o mesmo índice. Essa estratégia permite lidar de forma eficiente com colisões, mantendo o custo das operações próximo a O(1) na média.

Funções Hash Implementadas

O sistema permite alternar dinamicamente entre diferentes funções de dispersão, com rehash automático sempre que ocorre a troca. As funções implementadas foram:

- **Método da Divisão:** aplica o operador módulo sobre o valor numérico derivado da chave. É simples e rápido.
- **Método da Multiplicação:** utiliza uma constante fracionária para gerar o índice, garantindo uma dispersão mais uniforme dos valores.
- **Método da Dobra (folding):** proposto como melhoria, divide o código alfanumérico em partes, soma seus valores numéricos e aplica o módulo. Esse método mostrou-se mais adaptado para chaves mistas (letras e números).

Tratamento de Colisões

O tratamento de colisões foi realizado por meio de encadeamento separado, em que cada bucket da tabela hash armazena uma lista ligada de peças. Sempre que dois ou mais códigos são mapeados para o mesmo índice, eles passam a ocupar a mesma lista, preservando a integridade dos dados e garantindo que nenhuma peça seja sobrescrita indevidamente.

Essa abordagem foi escolhida por sua simplicidade de implementação e por manter as operações de inserção, busca e remoção com custo médio próximo a O(1), mesmo em situações de colisão. Além disso, a análise das listas em cada bucket fornece estatísticas importantes para avaliar a eficiência da função hash escolhida.

Testes e resultados:

Para validar a implementação, foram realizados testes de inserção, busca, remoção e carregamento de dados a partir de arquivos CSV. O sistema se mostrou funcional em todas as operações.Durante os testes iniciais, com a inserção de 78 registros, a tabela apresentou os seguintes resultados:

OBS: O tamanho da tabela já foi alterado automaticamente dobrando e encontrando o próximo número primo já que para todas as funções ao inserir essa quantidade de registros em uma tabela hash de 101 posições o fator de carga ultrapassou os 0.75, sendo obrigatório o Rehash da tabela.

Método da Multiplicação:

- **Tamanho da tabela (m):** 211
- **Número de itens (n):** 78
- **Fator de carga (α):** 0,37
- **Buckets utilizados:** 64 (30,33%)
- **Comprimento da maior lista (colisões):** 3

Método da Divisão:

- **Tamanho da tabela (m):** 211
- **Número de itens (n):** 78
- **Fator de carga (α):** 0,37
- **Buckets utilizados:** 67 (31,75%)
- **Comprimento da maior lista (colisões):** 2

Método da Divisão:

- **Tamanho da tabela (m):** 211
- **Número de itens (n):** 78
- **Fator de carga (α):** 0,37
- **Buckets utilizados:** 54 (25,59%)
- **Comprimento da maior lista (colisões):** 4

Com base nesses resultados, observou-se que a função Divisão obteve melhor dispersão dos elementos, reduzindo o número de colisões e tornando as buscas mais eficientes.

Também foram realizados testes com arquivos CSV, confirmando o correto carregamento e salvamento das peças.