Hash Table

SERGIO SIVONEI DE SANT'ANA FILHO GRR20242337
EDUARDO KALUF GRR20241770

June 22, 2025

1 Apresentação

O objetivo deste trabalho é implementar a inclusão e exclusão de valores em uma tabela hash de endereçamento aberto. Teremos duas tabelas hash de tamanhos iguais (11) e duas funções hash, a primeira será $h_1(k) = k \bmod m$ e a segunda é $h_2(k) = \lfloor m \cdot (k \cdot 0.9 - \lfloor k \cdot 0.9 \rfloor) \rfloor$.

Nós trataremos colisão movendo as chaves entre as duas tabelas. Basicamente simularemos o algoritmo Cuckoo Hash, porém sem nos preocuparmos com rehashing.

2 Implementação

2.1 Inclusão

A inclusão sempre será em T1, caso a posição calculada pela função hash estiver vazia ou marcada como "excluída", iremos inserir a chave k.

Caso exista colisão em T1 devemos realizar outras duas operações:

- Copiaremos a chave que está em T1 para T2. Incluindo-a em T2 usando h2("chave_antiga").
- Incluiremos a nova chave em T1 na posição h1("chave_nova").

A estrutura irá ignorar chaves duplicadas, neste caso simplesmente mantemos a primeira chave inserida.

2.2 Busca

Se a posição da chave k calculada por h1(k) estiver vazia, iremos retornar null, ou seja, a chave não existe.

Se a chave k existe em T1, retornaremos a posição dela em T1 calculada por h1(k). Caso contrário, a chave k existe em T2 e retornaremos a posição dela em T2 com h2(k).

2.3 Exclusão

Podemos neste trabalho excluir qualquer chave de qualquer tabela. Considere nesta exemplicação que a chave a ser excluída k existe em T1 ou T2.

Se a chave a ser excluida estiver na posição calculada por h2(k) em T2, então apenas excluimos a chave de T2.

Caso a chave esteja em T1 na posição calculada por h1(k), marcaremos a posição como "excluida", para que ainda existam meios de acessar as chaves de T2 com a busca, em seguida podemos excluir com segurança a chave k.

3 Diretórios e Arquivos

- `hashtable.h`
  Arquivo header que contém a documentação de cada função utilizada.
- `hashtable.c`
  Arquivo de implementação das funções da HashTable.
- `main.c`
  Arquivo responsável por receber os comandos passados pelo terminal e chamar as funções correspondentes, gerenciando o fluxo de dados do trabalho.
- `Makefile`
  Arquivo simples utilizado para compilar e fazer a linkagem do trabalho.
  - `make` → gera o executável `myht`
  - `make clean` → remove os arquivos gerados pela compilação
- `testes` e `testes.sh`
  `testes.sh` é um script simples que verifica, através dos testes e soluções armazenados no diretório `testes`, se a HashTable implementada está apresentando o comportamento esperado.
  O script compila automaticamente o projeto e o executa utilizando o comando `diff`.

  Sendo assim, caso nada seja impresso no terminal, todos os testes foram bem-sucedidos.

4 Conclusão

Ao longo deste estudo, aprofundamos nosso entendimento sobre a implementação e o funcionamento da HashTable. Ficou evidente que sua principal vantagem reside na excepcional eficiência com que executa as operações de inserção, remoção e busca de elementos. A versatilidade de sua aplicação, que vai desde a otimização de consultas em bancos de dados até o gerenciamento seguro de senhas, reforça ainda mais o valor desta estrutura de dados.