

Estrutura de Dados II



Equipe: Debbugs on the Table

Sumário

- Tabela Hash: Como funciona e para que serve?
- Nosso Programa.
- Como Desenvolvemos.
- Dificuldades Encontradas.
- Benefícios do Nosso Código

Tabela Hash

É uma estrutura que associa chaves a valores (como um dicionário). Internamente, ela usa uma função hash para transformar a chave em um índice de um array, onde o valor será armazenado.

Qual problema ela resolve?

- Busca lenta em listas ou arrays ($O(n)$)
- Inserções e remoções ineficientes
- Organização e associação de dados via chave/valor (como "CPF → nome")

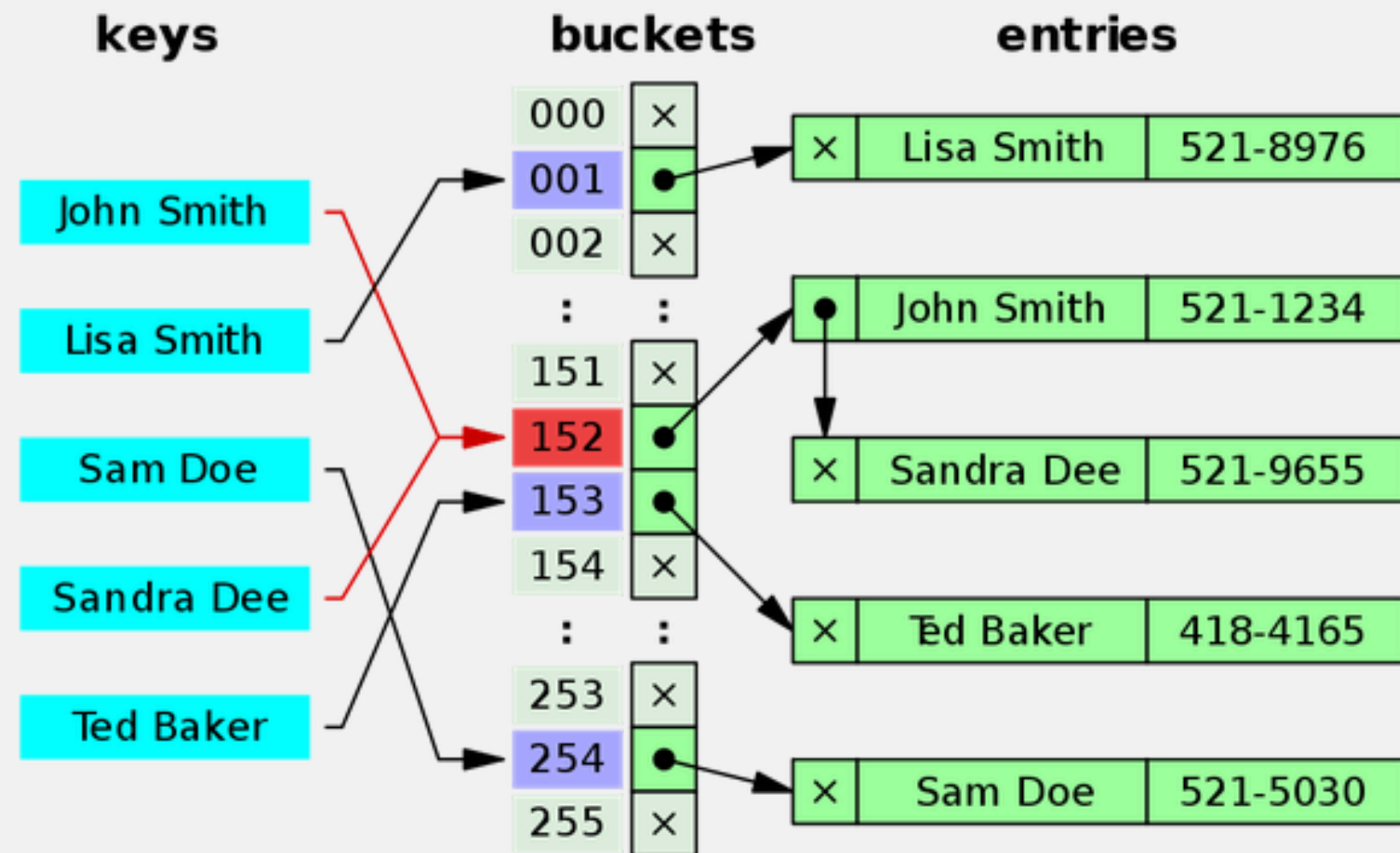
Vantagens

- Tempo médio de acesso: $O(1)$
 - constante
- Ideal para operações como:
- Verificar se um item existe
- Associar dados a identificadores únicos (ex: nomes de usuário, CPF, etc.)

Cuidados

- Colisões
- Escolha da função hash é crucial para bom desempenho.

Exemplos



pseudocodigo

```
funcao hash(key, T):
```

```
    A = 0.6180339887 // constante
```

```
    valor = key * A
```

```
    parte_fracionaria = valor - floor(valor)
```

```
    indice = floor(T * parte_fracionaria)
```

```
    retorna indice
```

Desenvolvimento

Passo 1 Criação hash.h

Criação de pré-funções e definições de constantes

Passo 2 Desenvolvimento da hash.c

Criação da tabelas, funções e switches

Passo 3 Melhorias

Tratamento de erros e inicialização do flask

Passo 4 html/css

Criação das páginas e estilização

Passo 5 Integração final

Link entre os arquivos

Demonstrações: Nosso Programa

Dificuldades Encontradas

Desafios

- Tratamento de colisões, quando duas chaves;
- Transformar String é um inteiro;
- Tratar erro que não permitia encontrar chaves na tabela;
- Falha na liberação de memória ao remover uma chave.

Soluções

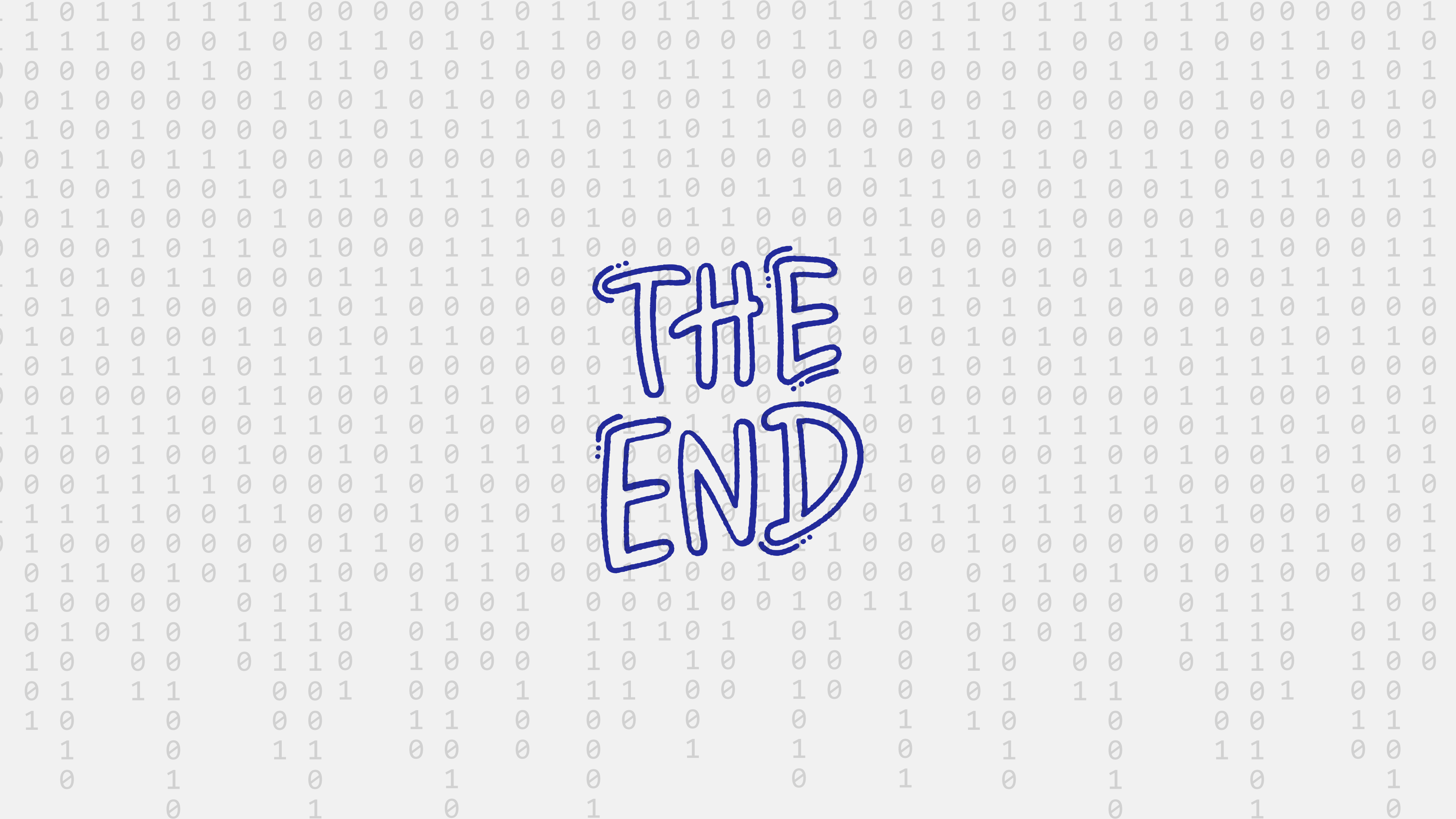
- Uso de listas encadeadas para solucionar colisões;
- Uso do método da multiplicação para descobrir o índice da chave;
- Uso da função nativa strcmp() para comparar de forma correta as chaves;
- Atualiza os ponteiros para remover o nó e só depois chama free(atual).

Benefícios do Nosso Código

- Uso modularização que permite desenvolver um código organizado e objetivo;
- Implementação do método da multiplicação para descobrir o índice em que será inserido a chave;
- Implementação de uma interface visando auxiliar na compreensão do funcionamento de uma tabela hash. Além disso, a interface auxilia a representar o conceito o qual aplicamos essa estrutura de dado.

Melhorias Futuras

- Aprimorar segurança utilizando uma interface de login e senha;
- Implementação de um banco de dados com MySQL para armazenar os dados gerados pelos usuários;
- Aprimorar interface para melhor interação com cliente/empresa;
- Melhorar o tratamento de erro da tabela hash.



THE
END