

# TABELAS HASH

## DCE792 - AEDs II (Prática)

Atualizado em: 20 de novembro de 2025

Iago Carvalho

Departamento de Ciência da Computação



# TABELA HASH

Tabelas hash também são conhecidas como tabelas de espalhamento, tabelas de indexação ou mapas

Uma tabela hash é uma estrutura de dados que possibilita o acesso a seus registros de forma imediata

- $O(1)$

Além disso, a inserção e remoção de registros também de forma imediata -  $O(1)$

Claro, tudo isto na teoria

- Na prática, pode haver diferenças

# TABELA HASH

Uma tabela hash pode ser representada computacionalmente como um vetor

- Necessariamente, é um vetor contíguo
- Não é possível implementar utilizando apontadores

Este vetor tem um tamanho fixo  $M$

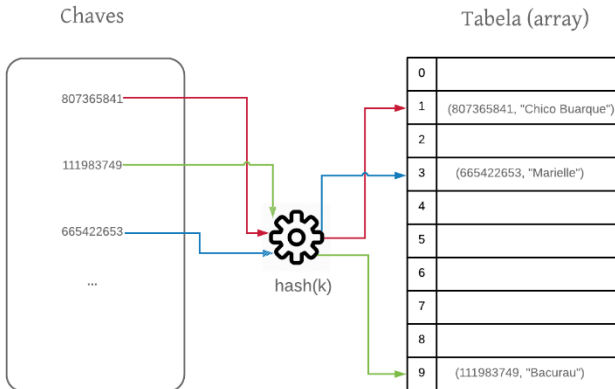
- Desta forma, é possível armazenar um total de  $M$  chaves
- Custo de memória fixo igual a  $M$  vezes o tamanho da *struct* a ser armazenada

O acesso, inserção e remoção de itens em uma tabela hash é realizada através de uma **função de hashing**

# FUNÇÃO DE HASHING

Uma função de hashing  $f(k) \mapsto y : y \in [0, |M|]$  mapeia uma chave  $k$  em uma posição  $y$  da tabela de hash

- Função de hashing também é conhecida como *função de espalhamento*



# EXEMPLOS DE FUNÇÃO DE HASHING

Uma função de hashing é função matemática **sobrejetora**

- Funções nas quais todo elemento do contradomínio ( $y$ ) é imagem de pelo menos um elemento do domínio ( $k$ )

Exemplos de função de hashing

- $y = k \bmod M$
- $y = k^2 \bmod M$
- $y = 2k + 5 \bmod M$

Pela propriedade de sobrejetora, pode-se perceber que existe a possibilidade de que duas chaves diferentes  $i$  e  $j$  sejam posicionadas na mesma posição  $k$

Seja  $i = 3$ ,  $j = 8$ ,  $M = 5$  e  $f(k) = k \bmod M$

- Neste caso, tanto a chave  $i$  como a chave  $j$  seriam posicionados na terceira posição da tabela hash
  - $f(3) = 3$
  - $f(8) = 3$

Como resolver?

Existem dois métodos clássicos para tratar colisões em tabelas hash

1. Uso de listas encadeadas (*separate chaining*)
2. Endereçamento aberto

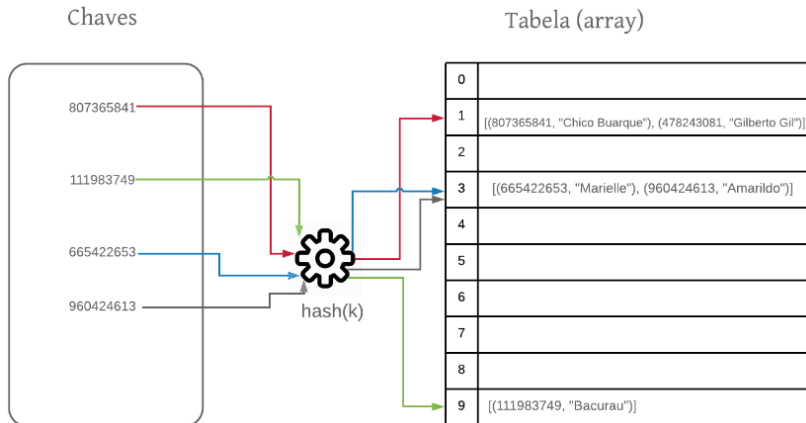
Nesta aula, estudaremos tabelas hash implementadas utilizando listas encadeadas

- Na próxima aula, estudaremos o funcionamento do endereçamento aberto

# RESOLUÇÃO DE COLISÕES POR ENCADEAMENTO

Cada posição da tabela hash é uma lista encadeada

- Caso a posição já esteja ocupada, a inserção é realizada no fim da lista





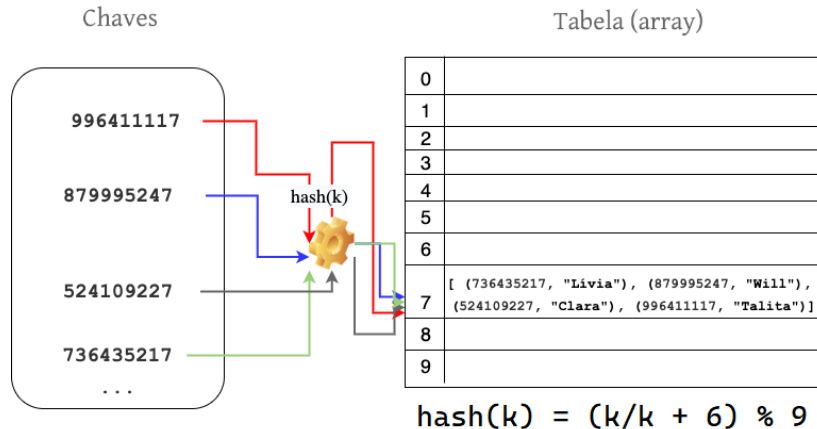
# VANTAGENS E DESVANTAGENS

O uso de listas encadeadas permite armazenar um número de chaves maior do que  $M$

O custo de inserção, remoção e acesso é modificado

- No pior caso, é linear no número de chaves inseridas
- Caso extremamente particular onde todas as chaves são mapeadas para uma mesma posição

# PIOR CASO



# ATIVIDADE PRÁTICA

Implementar uma hash com este método de resolução de colisão

O código base está disponível no [Github](#)

- Ele implementa uma função hash  $f(k) = k \% M$
- Possui funções de inserção, pesquisa e remoção
- Não possui tratamento de colisão

PRÓXIMA AULA:

TABELAS HASH COM  
ENDEREÇAMENTO ABERTO