

Tabela Hash

Disciplina: Estruturas de Dados

Professora: Camila Taumaturgo

e-mail: camila.taumaturgo@ifrn.edu.br



Sumário

- Tabela Hash
 - Introdução
 - Princípios de funcionamento
 - Funções de Dispersão
 - Tratamento de colisões por encadeamento

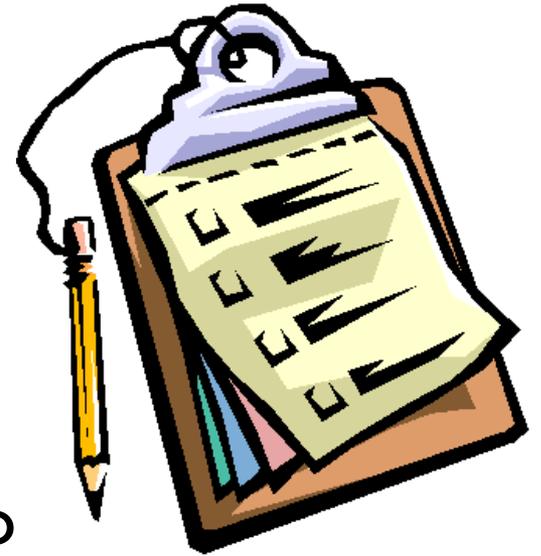


Tabela Hash

- Uma Tabela Hash, também conhecida como tabela de dispersão ou tabela de espalhamento, é uma estrutura de dados especial, que associa chaves e valores.
- Seu objetivo é a partir de uma chave simples, fazer uma busca rápida e obter o valor desejado.



Introdução

- A Tabela Hash leva em conta o valor absoluto de cada chave, interpretado como um valor numérico.
- Através da aplicação de uma função conveniente, a chave é transformada em um endereço de uma tabela



Introdução



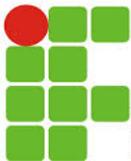
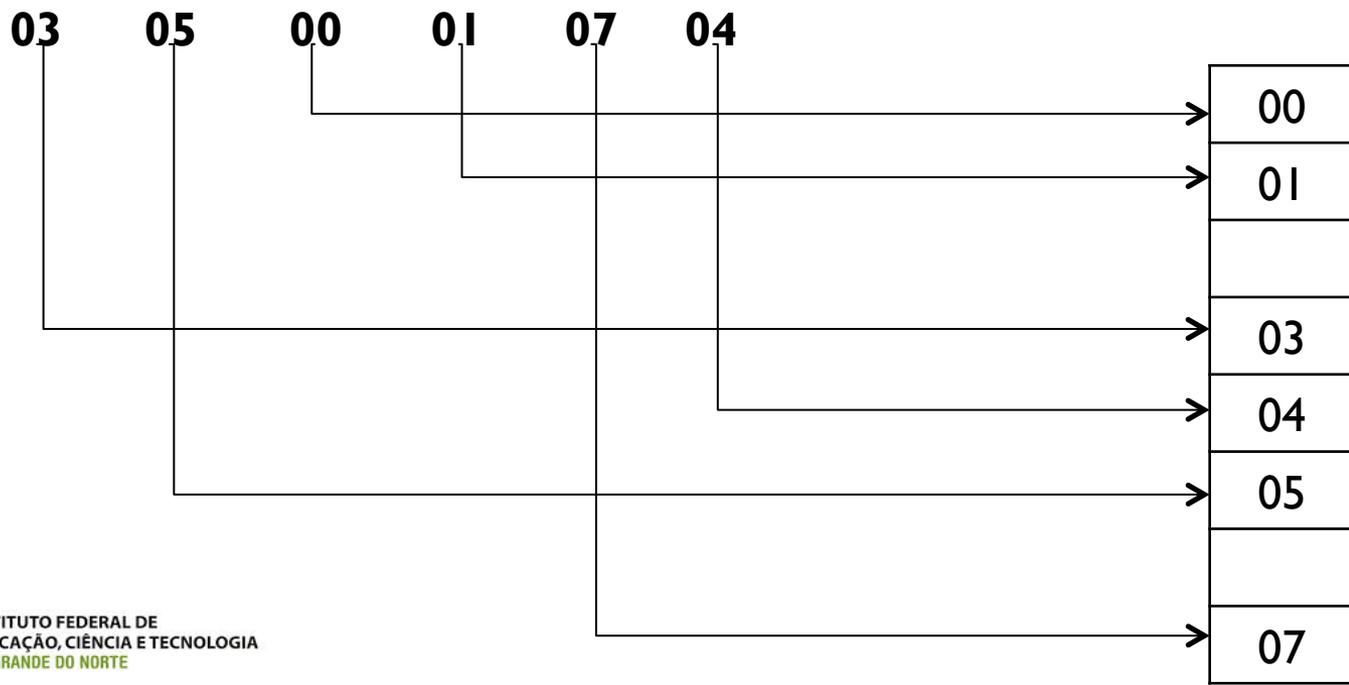
Princípios de Funcionamento

- Suponha que existam n chaves a serem armazenadas em uma tabela T , sequencial e de tamanho m .
- As posições da tabela se situam no intervalo $[0, m-1]$.
- Isto é, a tabela é particionada em m *compartimentos*, cada uma corresponde a um endereço e podendo armazenar r nós distintos.



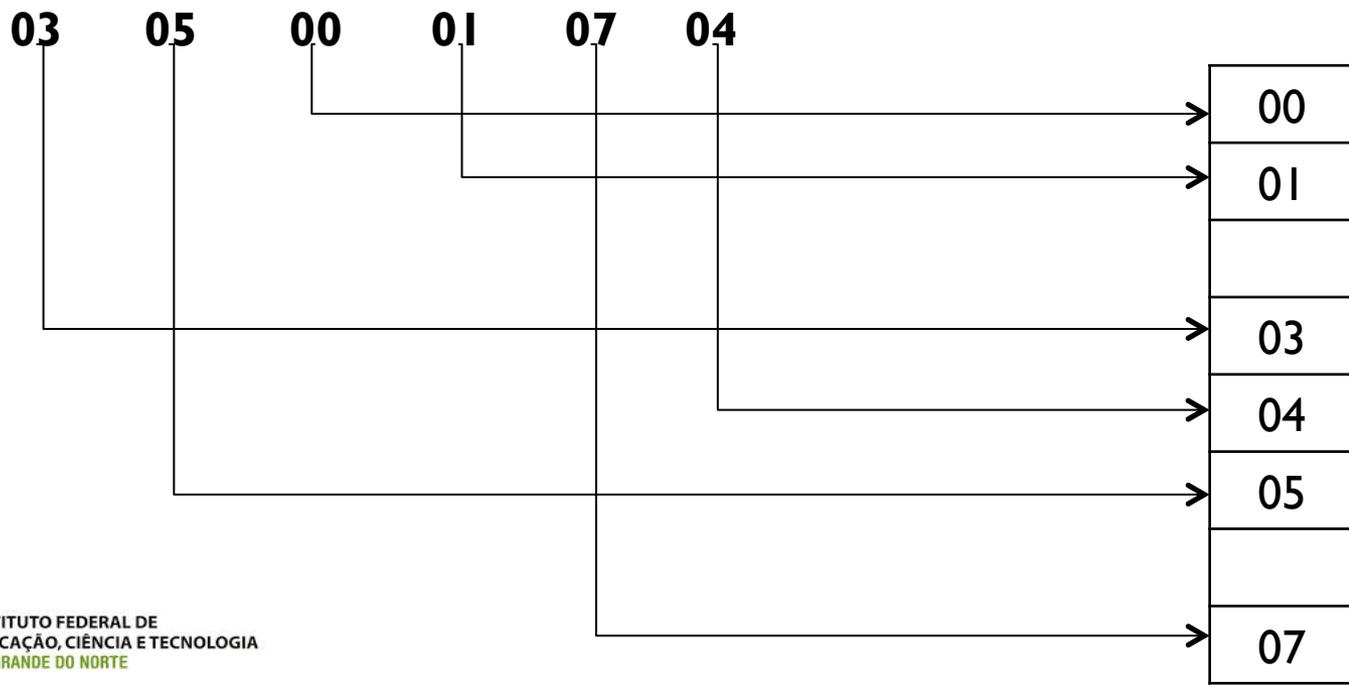
Princípios de Funcionamento

- O objetivo é armazenar cada chave no bloco referente ao seu endereço.
- A busca, assim, requer somente um acesso a um bloco.



Princípios de Funcionamento

- Valor da chave como seu índice na tabela.
- Cada chave x é adicionada no compartimento x .
- Acesso direto.



Princípios de Funcionamento

- Valor da chave como seu índice na tabela.
- Cada chave x é adicionada no compartimento x .
- Acesso direto.
- O acesso direto pode ser usado no caso $n < m$.



Princípios de Funcionamento

- Chaves nem sempre são valores numéricos.
- As chaves podem consistir em nomes de pessoas.
- Solução: Todo dado não numérico corresponde uma representação numérica no computador.
- Assim sendo, todas as chaves são consideradas numéricas.



Princípios de Funcionamento

- Vamos utilizar sempre o acesso direto?

Não

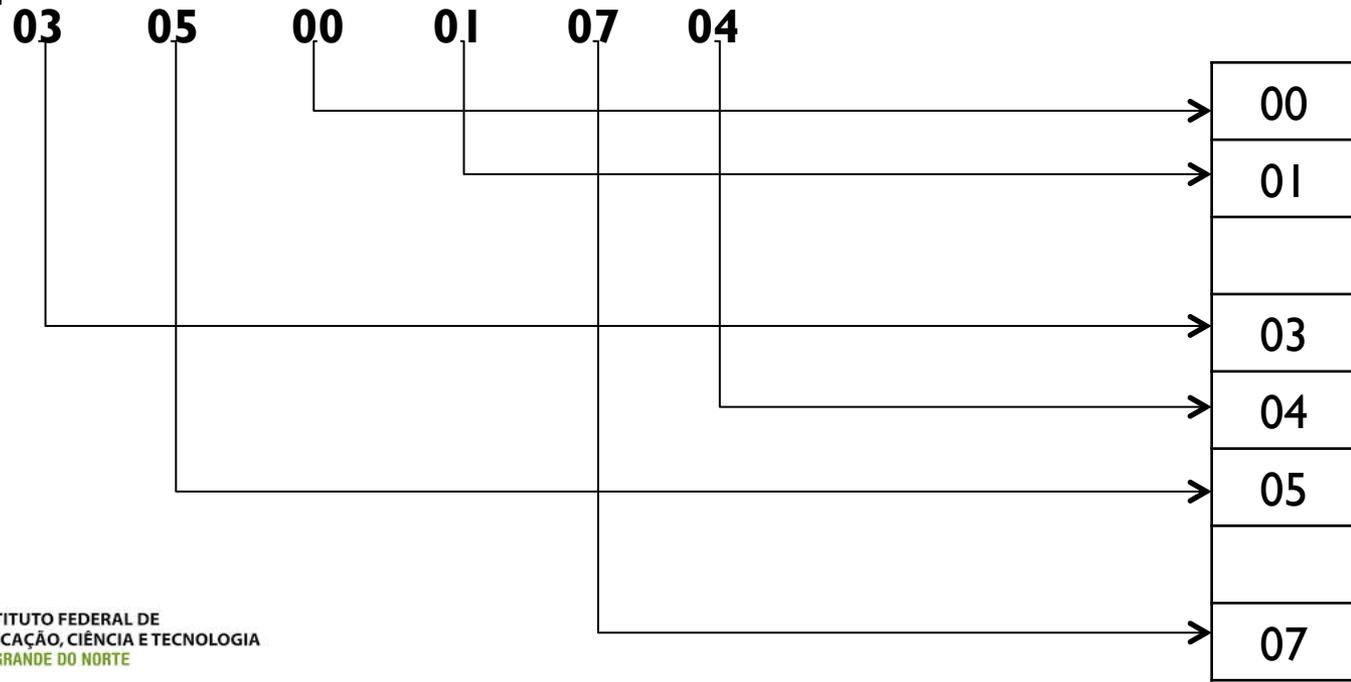


Princípios de Funcionamento

- Vamos utilizar sempre o acesso direto?

Não

- A quantidade de espaços vazios pode ser proibitiva.



Princípios de Funcionamento

- Como resolver?



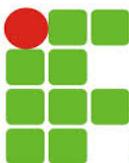
Princípios de Funcionamento

- Como resolver?
 - Transformar cada chave x em um valor no intervalo $[0, m-1]$.
 - Através de uma *função de dispersão* h .
 - Dada a chave x , determina-se o valor $h(x)$, denominado *endereço-base* de c .
 - Se o compartimento $h(x)$ estiver desocupado, poderá ser utilizado para armazenar a x .



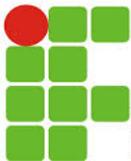
Princípios de Funcionamento

- A função de dispersão pode não garantir a injetividade.
- É possível a existência de outra chave $y \neq x$, tal que $h(y) = h(x)$.
- Esse fenômeno é denominado de *colisão*.
- As chaves x e y são sinônimas em relação a h .
- Na ocorrência desse fato, utiliza-se um procedimento especial, denominado *tratamento de colisão*.



Função de dispersão

- Uma função de dispersão h transforma uma chave x em um endereço-base $h(x)$ da tabela de dispersão.
- Idealmente, uma função de dispersão deve satisfazer às seguintes condições:
 - produzir um número baixo de colisão;
 - ser facilmente computável;
 - ser uniforme.



Função de dispersão

- Método da Divisão
- Método da Dobra
- Método da Multiplicação



Método da Divisão

- Fácil e eficiente
- Muito empregado
- A chave x é dividida pela dimensão da tabela m , e o resto da divisão é usado com endereço chave.

- Isto é

$$h(x) = x \text{ mod } m$$

- Resulta em endereços no intervalo



Método da Divisão

- Alguns valores de m são melhores que outros.
- Se m é um número par, $h(x)$ será par quando x for par e ímpar quando x for ímpar. **Não é uma boa solução.**
- Se m for uma potência de 2, $h(x)$ dependerá apenas de alguns dígitos de x . **A situação é pior.**



Método da Divisão

- Existem alguns critérios que têm sido aplicados com bons resultados práticos.
- Escolher m de modo que seja um número primo não próximo a uma potência de 2.
- Escolher m tal que não possua divisores primos menores que 20.

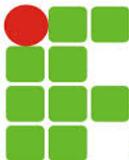
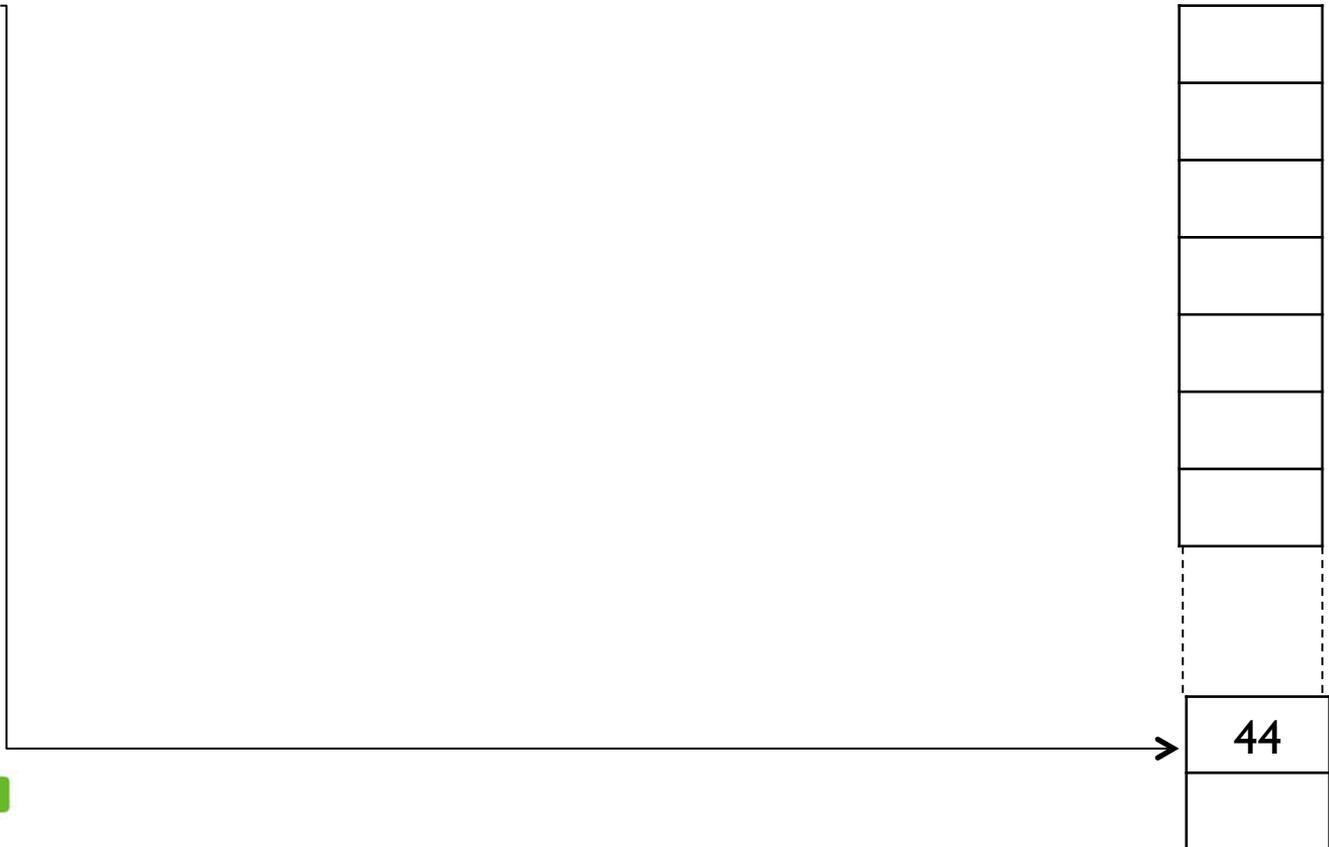


Método da Divisão

$$m = 23$$

$$44 \bmod 23 = 21$$

44 46 49 68 71 97

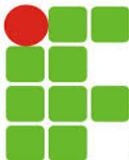


Método da Divisão

$$m = 23$$

$$46 \bmod 23 = 0$$

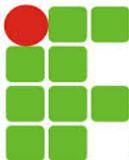
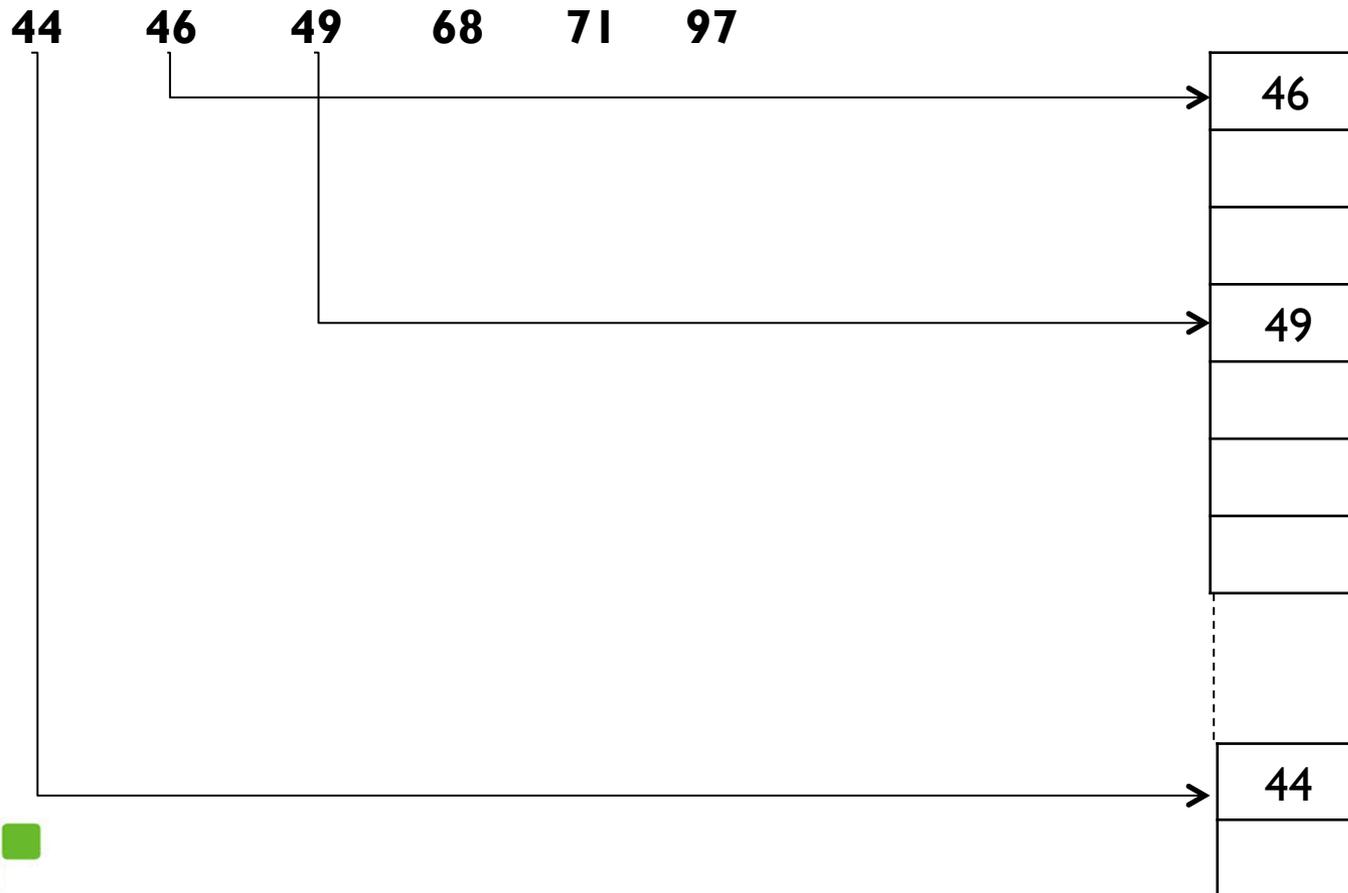
44 46 49 68 71 97



Método da Divisão

$$m = 23$$

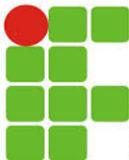
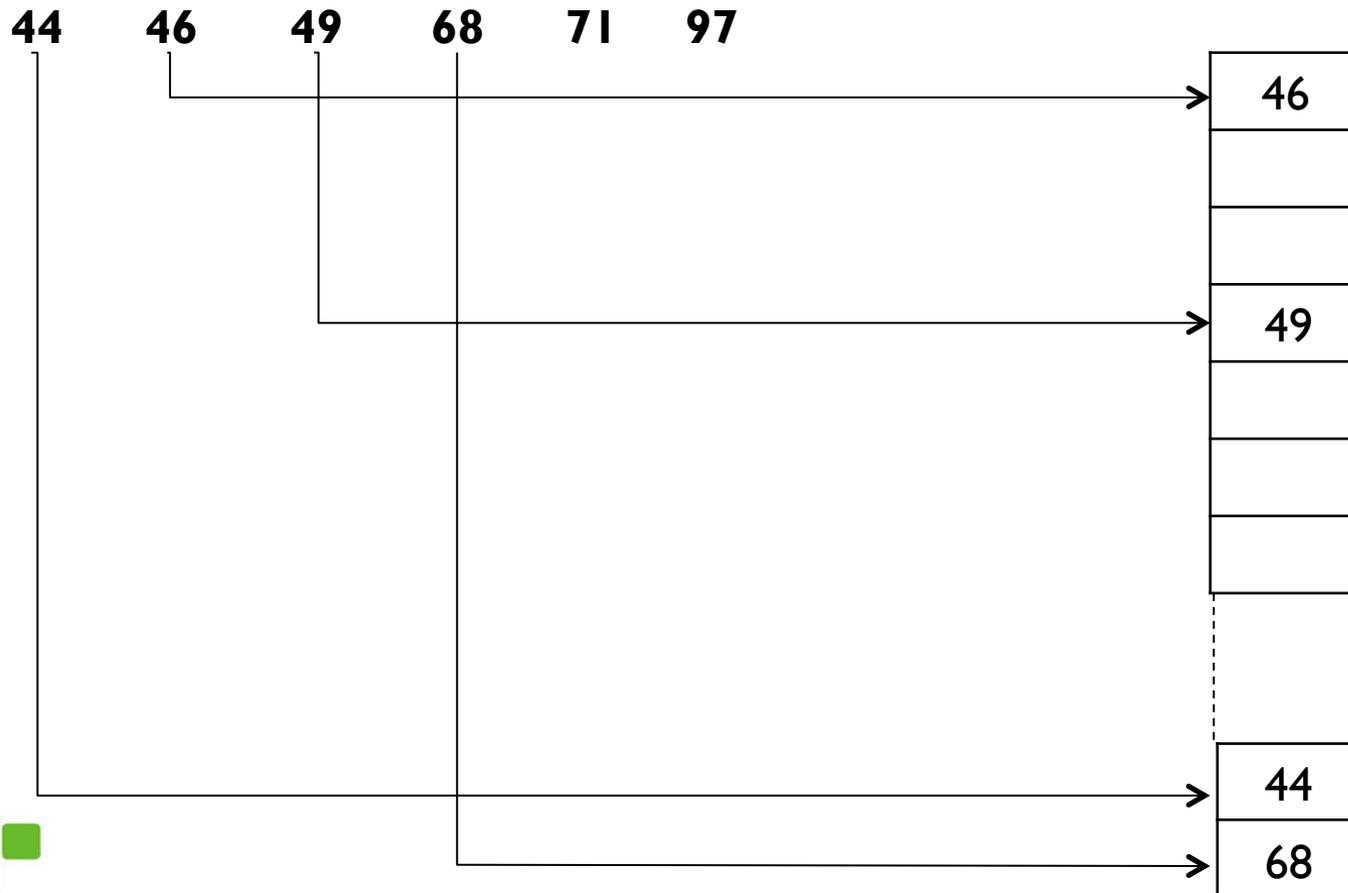
$$49 \bmod 23 = 3$$



Método da Divisão

$$m = 23$$

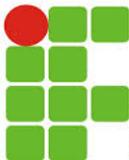
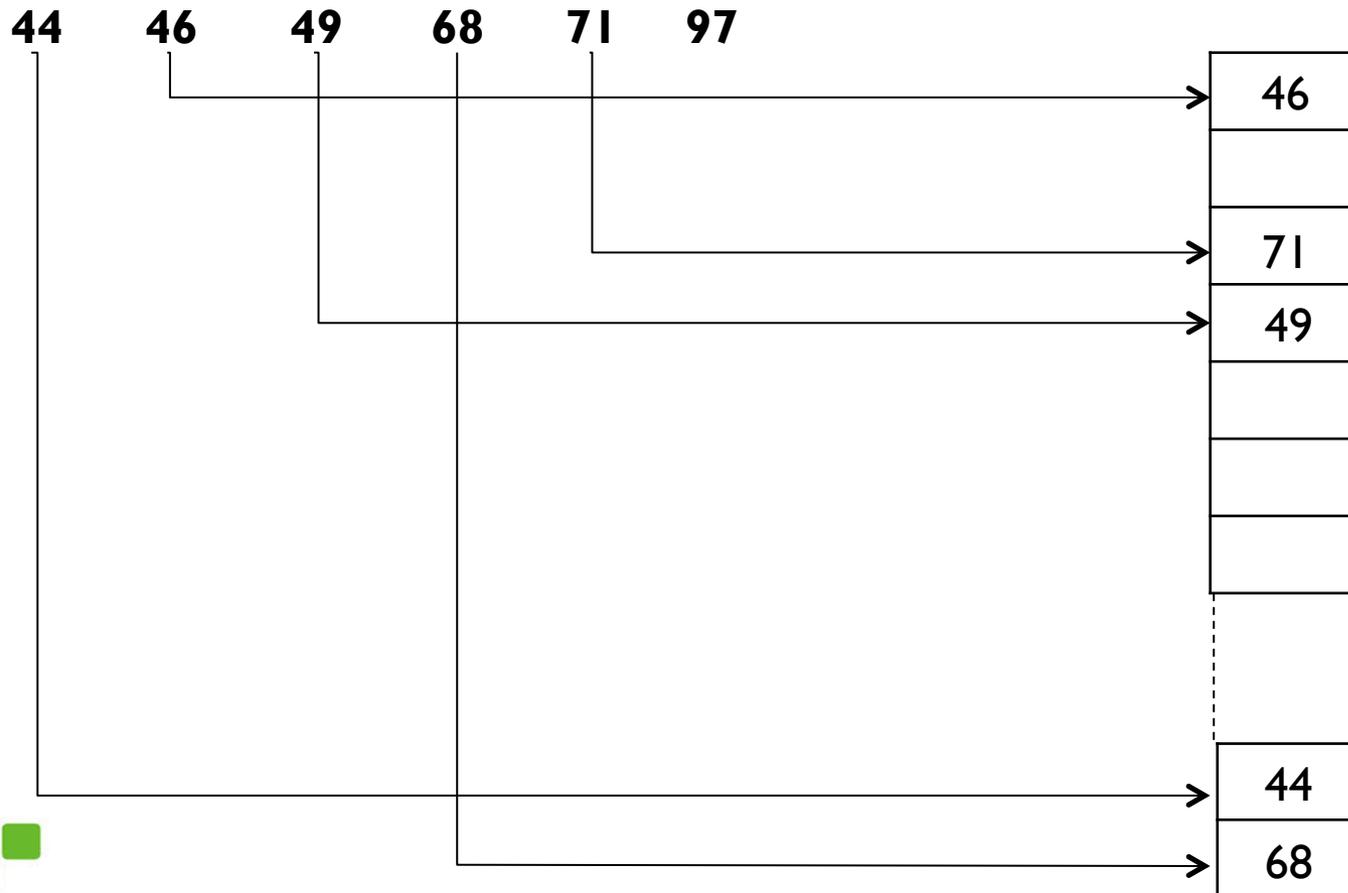
$$68 \bmod 23 = 22$$



Método da Divisão

$$m = 23$$

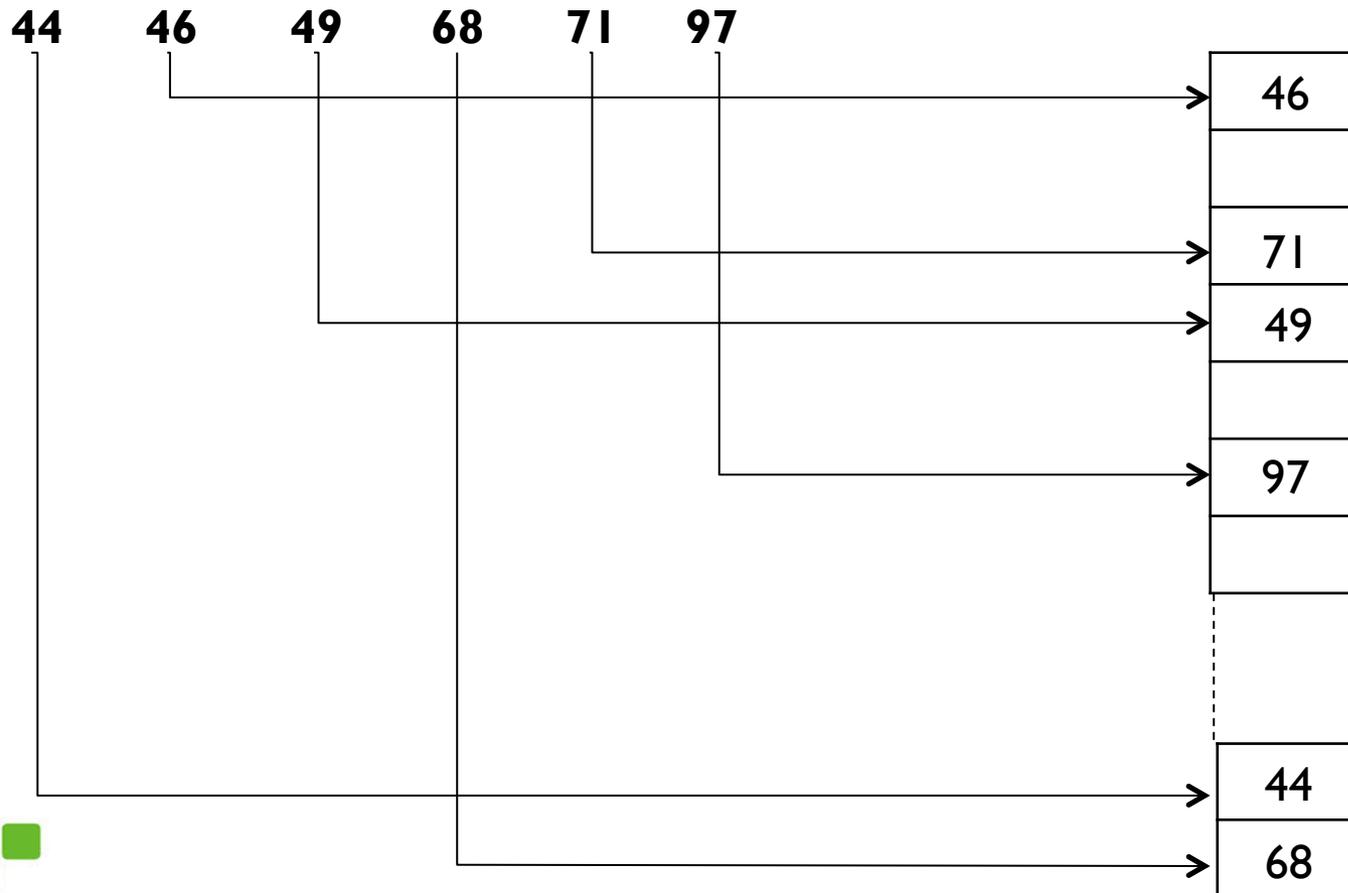
$$71 \bmod 23 = 2$$



Método da Divisão

$$m = 23$$

$$97 \bmod 23 = 5$$



Método da dobra

- Suponha a chave como uma sequência de dígitos escritos num pedaço de papel.
- O método em questão consiste em “dobrar” esse papel, de maneira que os dígitos se sobreponham.
- Esses devem ser somados sem levar em consideração o “vai um”.



Método da dobra

- Chave: 279384

$$t = 2$$

$$d = 6$$

- Método da Dobre

Dobra 27 em 93 e soma, descartando “vai um”.

Resultado Intermediário: 6584

Dobra 65 em 84 e soma ...

Resultado final: 30



Método da dobra

- Utilizando bits
- Chave: 71

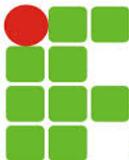
$$71 = 00010\ 00111$$

$$\begin{array}{r} 00010 \\ 00111 \\ \hline 00101 \end{array} \text{ ou exclusivo} = 5$$



Método da Multiplicação

- Existem algumas variações.
- Método mais conhecido: “Meio do quadrado”.
- A chave é multiplicada por ela mesma.
- O resultado é armazenada em palavra de memória de b bits.
- O número de bits necessário para forma o endereço-base de uma chave é então retirado dos b bits.
- Descartando-se os bits excessivos da extrema direita e da extrema esquerda.



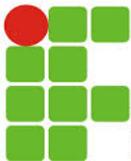
Tratamento de colisões por encadeamento

- O endereço base pode ser encontrado para chaves diferentes como resultado da função de dispersão.
- O que é chamado de colisão.
- Uma ideia para tratar colisões consiste em armazenar chaves sinônimas em listas encadeadas:
 - Encadeamento exterior
 - Encadeamento interior

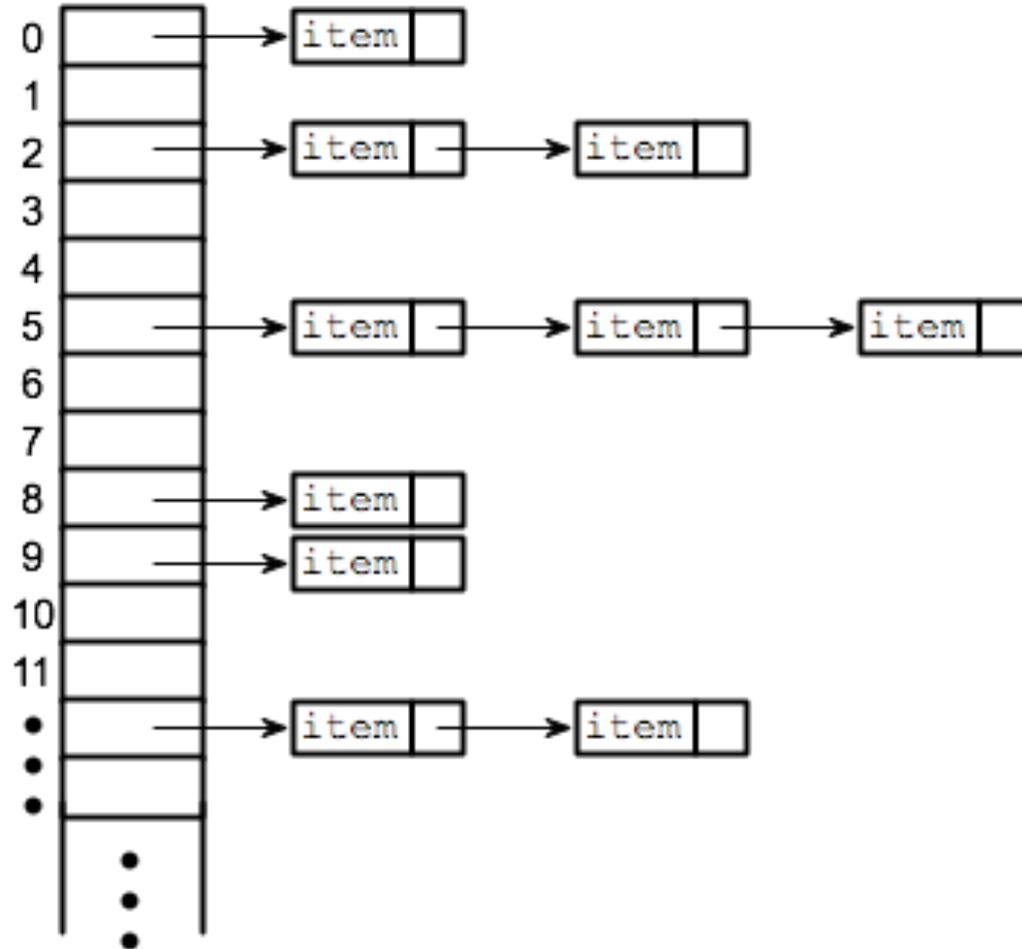


Encadeamento Exterior

- Solução muito usada,
- Consiste em manter m listas encadeadas, uma para cada possível endereço base.
- Um campo para o encadeamento deve ser acrescentado a cada nó.
- O nós correspondentes ao endereço-base serão apenas nós-cabeças para essas listas.



Encadeamento Exterior



Encadeamento Interior

- Não queremos adicionar m listas externas à tabela.
- Resolve-se o problema da colisão mediante o emprego de listas encadeadas, desde que estas compartilhem o mesmo espaço de memória que a tabela de dispersão.



Encadeamento Interior

- O encadeamento interior prevê a divisão da tabela T em duas zonas, uma de endereço-base, de tamanho p , e outra reservada aos sinônimos, de tamanho s .
- Naturalmente $p + s = m$
- Os valores p e s são fixos.
- A função de dispersão deve obter endereços-base na faixa $[0, p - 1]$ apenas.



Encadeamento Interior

- Cada nó da tabela possui dois campos, um para armazenar o valor, e outro um ponteiro que indica o próximo elemento da lista de sinônimos correspondentes ao endereço-base em questão.



Encadeamento Interior

- $n = 5 \quad 48 \quad 03 \quad 80 \quad 31 \quad 20$
- $m = 7$
- $p = 4$ e $s = 3$
- Função de dispersão: $h(x) = x \bmod 4$

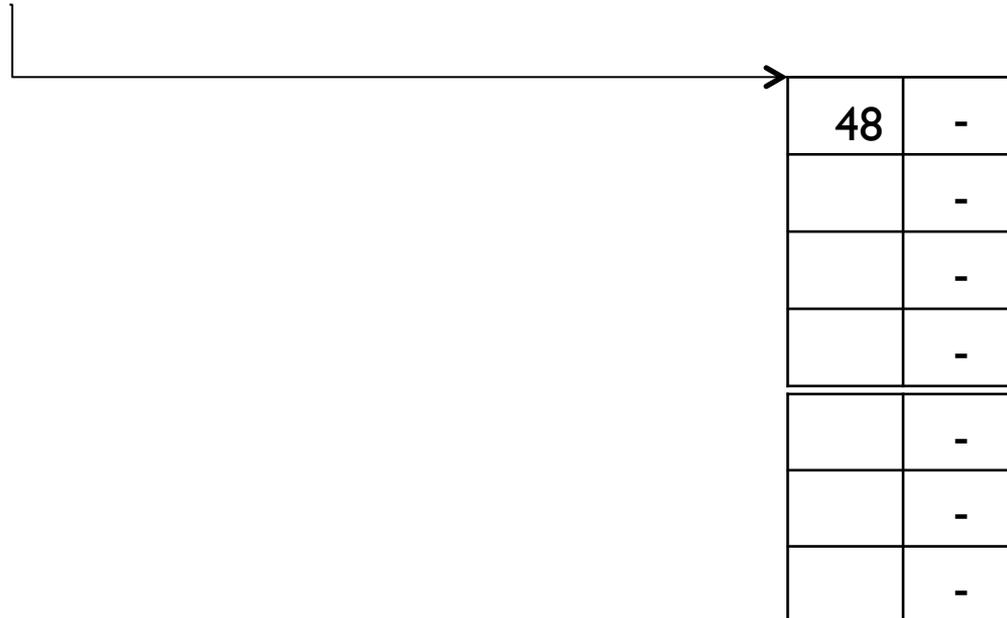


Encadeamento Interior

$$m = 7$$

$$48 \bmod 4 = 0$$

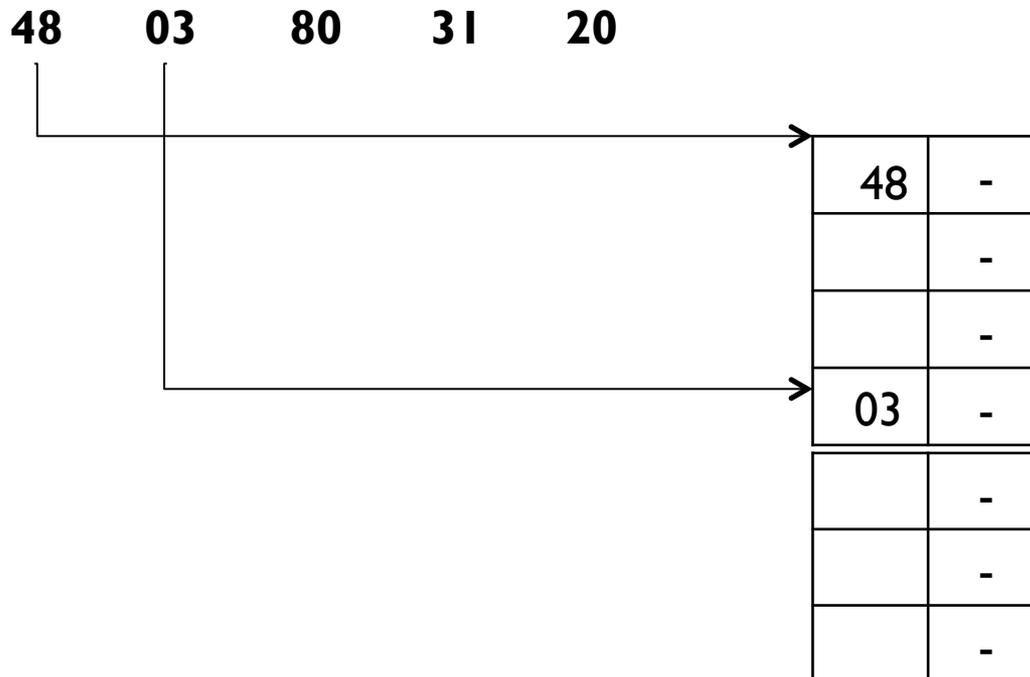
48 03 80 31 20



Encadeamento Interior

$$m = 7$$

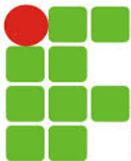
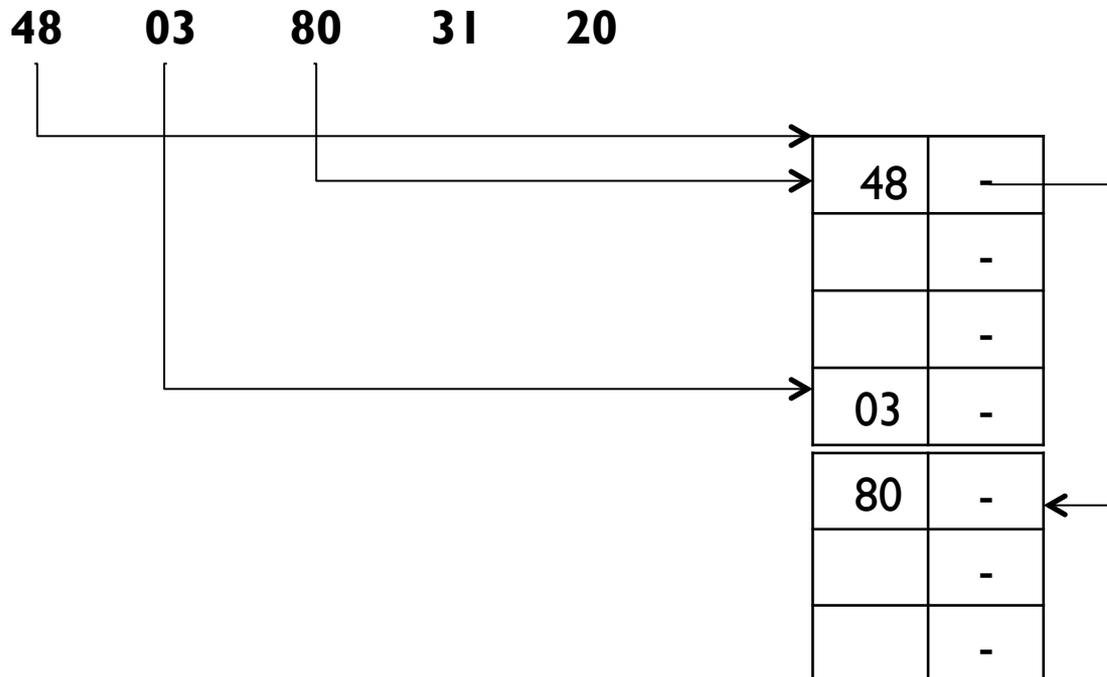
$$03 \bmod 4 = 3$$



Encadeamento Interior

$$m = 7$$

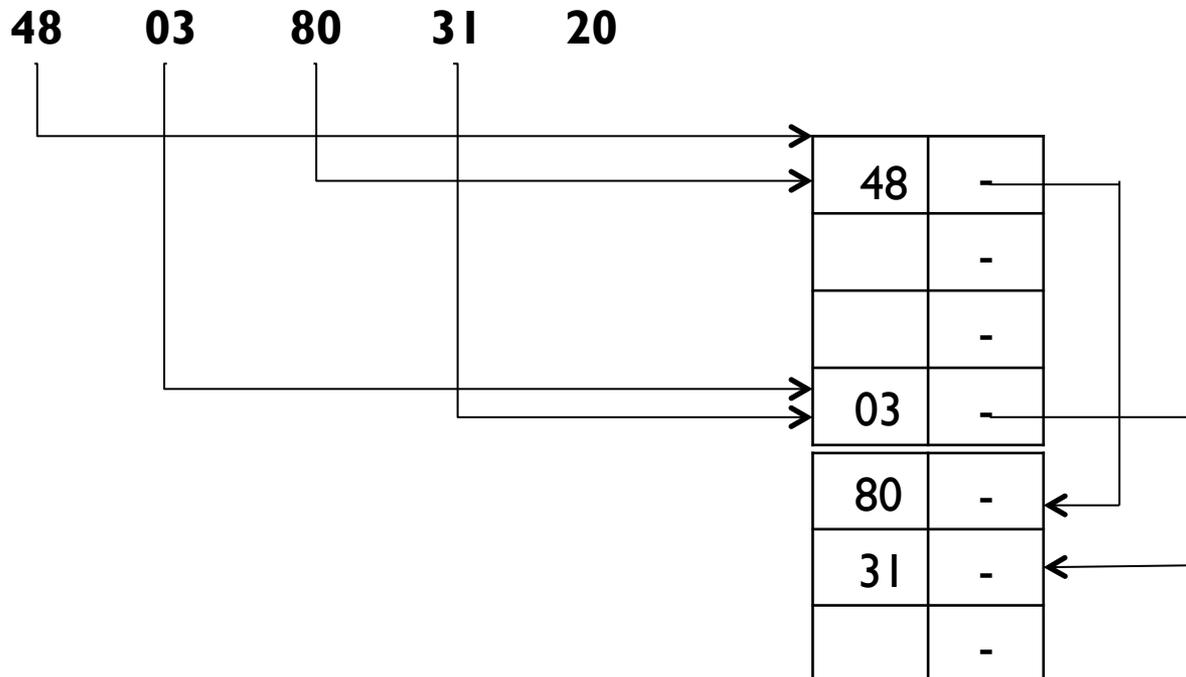
$$80 \bmod 4 = 0$$



Encadeamento Interior

$$m = 7$$

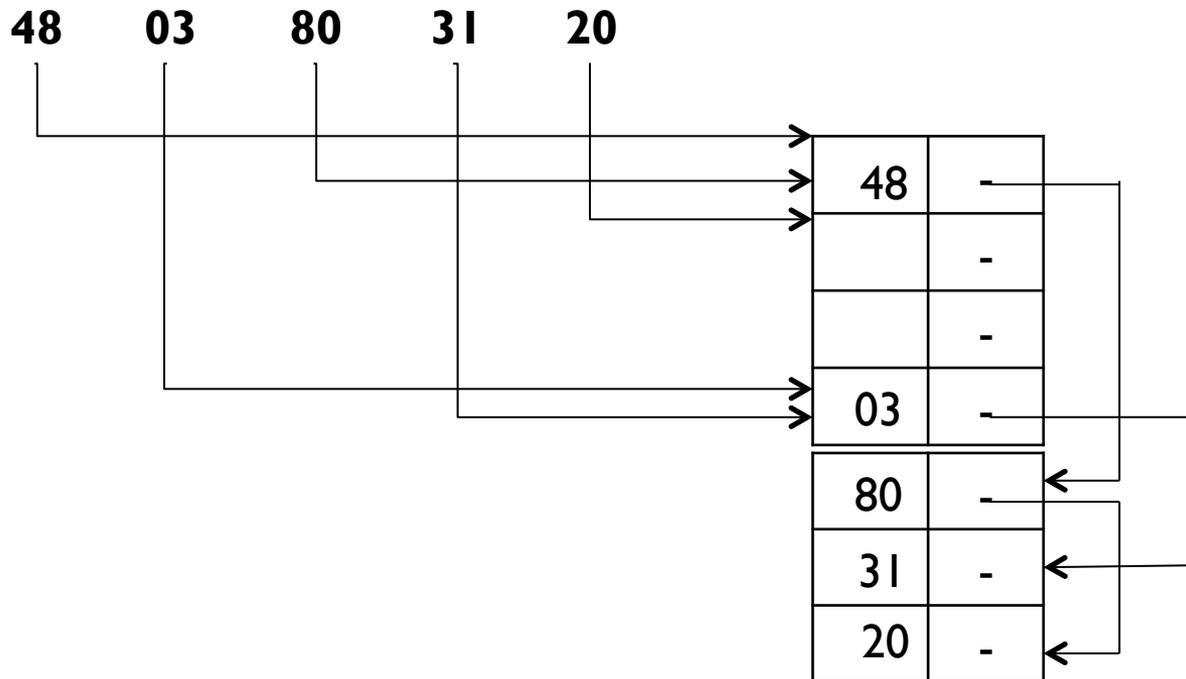
$$31 \bmod 4 = 3$$



Encadeamento Interior

$$m = 7$$

$$20 \bmod 4 = 0$$



Referências

- Goodrich, M. T., Tamassia, R. Estruturas de Dados e Algoritmos em Java, 2013.
- Nina Edelweiss e Renata Galante. Estruturas de dados, Série livros didáticos informática ufrgs 18, 2009.
- Szwarcfiter, J. L., Markenzon, L. Estruturas de Dados e Seus Algoritmos, 3 ed. 2010

