

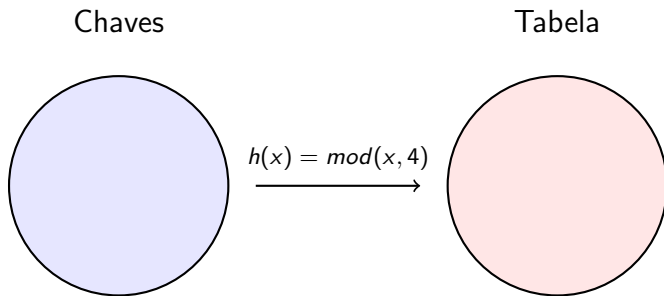
Tabela de Dispersão 2

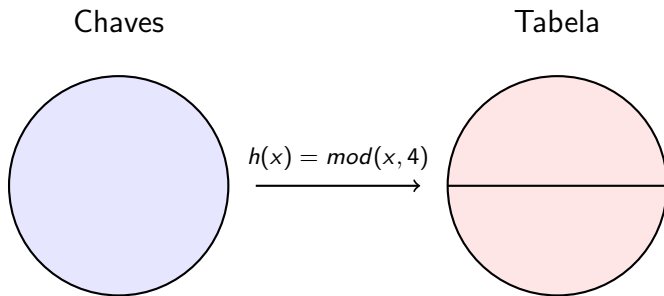
Funções Hash e tratamento de Colisões

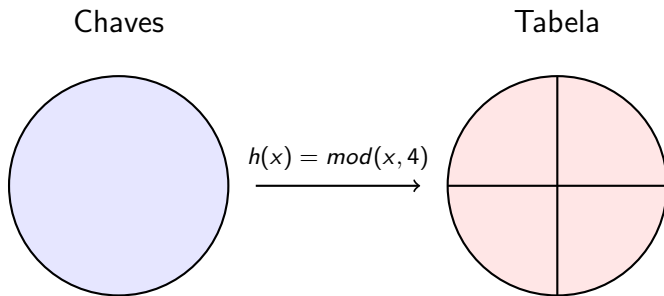
Prof. Kennedy Lopes

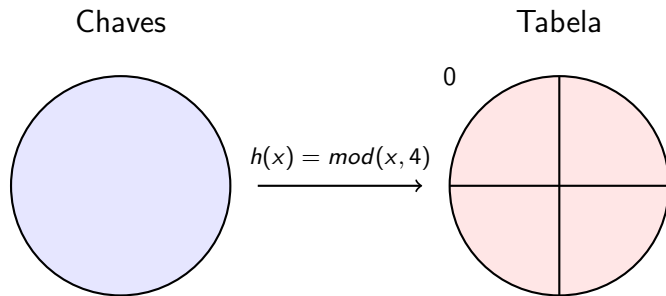
Universidade Federal do Semi-árido

31 de Agosto de 2021

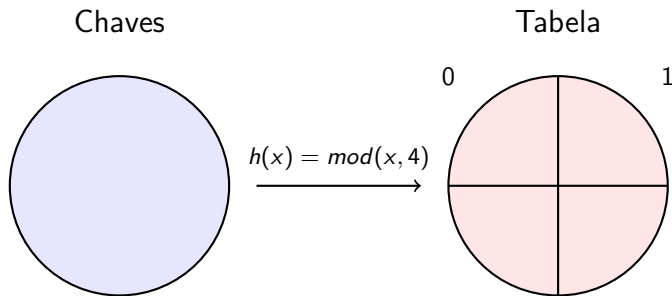




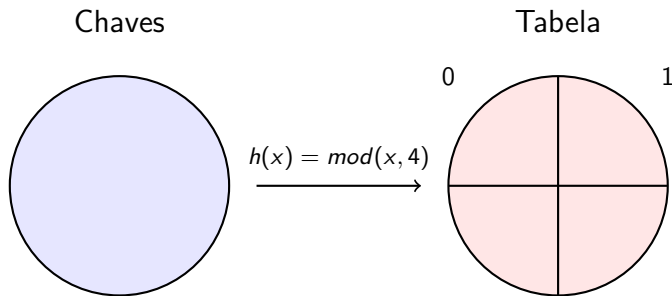




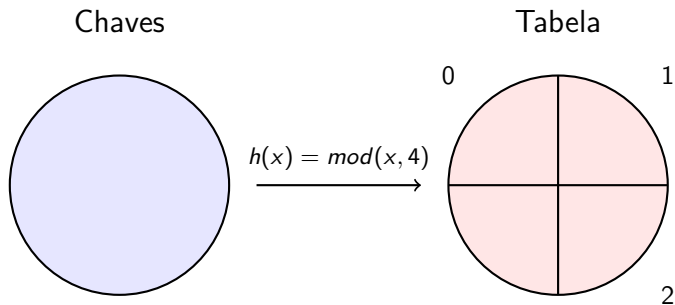
Funções Hash



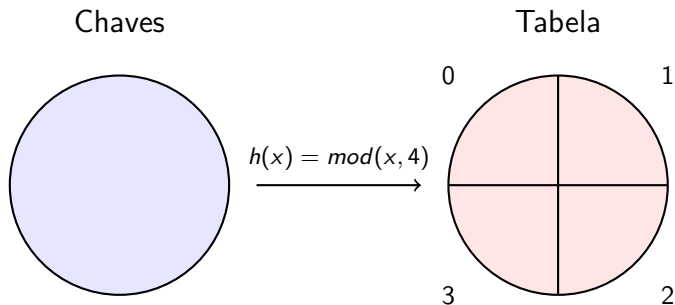
Funções Hash



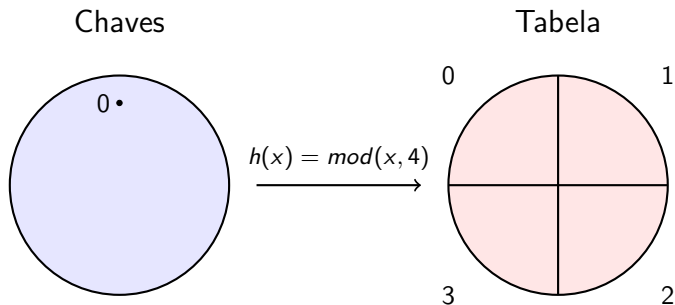
Funções Hash



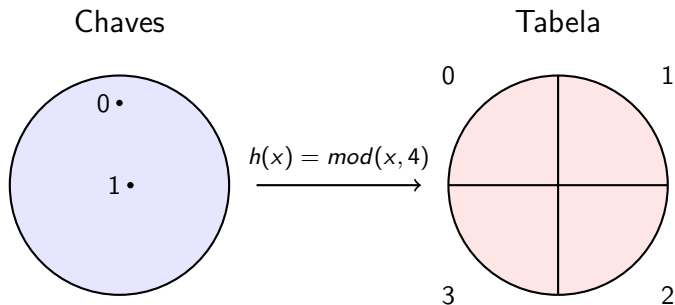
Funções Hash



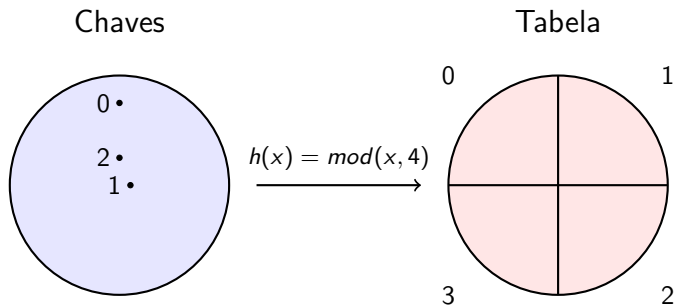
Funções Hash



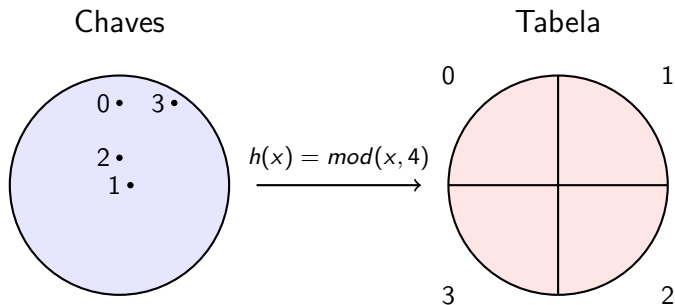
Funções Hash



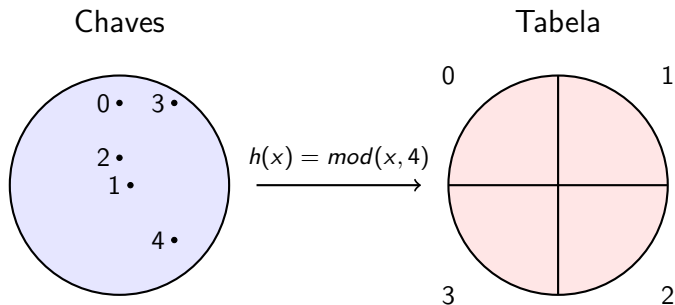
Funções Hash



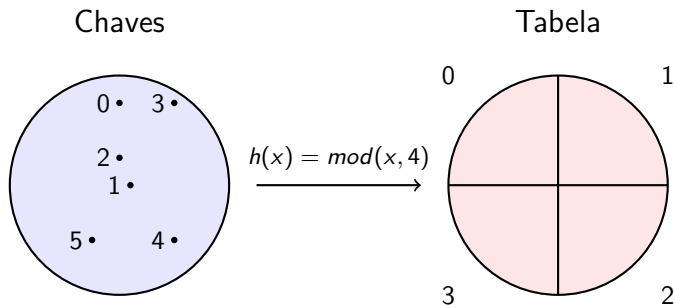
Funções Hash



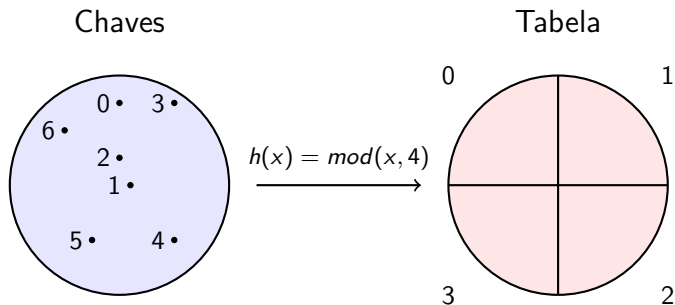
Funções Hash



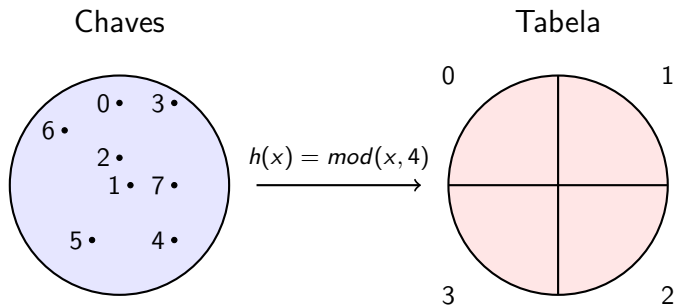
Funções Hash



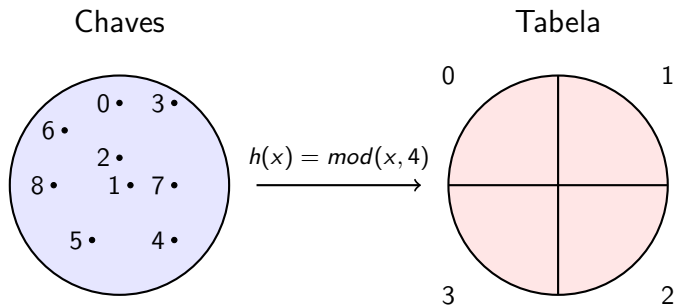
Funções Hash



Funções Hash



Funções Hash



Tratamento de Colisão

1. Mundo Ideal

- 1.1 Hashing perfeito! Função sempre fornecerá posições diferentes para cada chave inserida.

2. Mundo Real

- 2.1 Independente da função Hash, ocorrerá colisões. Inevitavelmente uma posição pode ter duas entradas.

Exemplo de Colisão

Paradoxo do aniversário:

Em um grupo com 23 pessoas juntas ao acaso, existe uma chance menor do que 50% de que duas pessoas comemorem aniversário no mesmo dia.

Exemplo de Colisão

Paradoxo do aniversário:

Em um grupo com 23 pessoas juntas ao acaso, existe uma chance menor do que 50% de que duas pessoas comemorem aniversário no mesmo dia.

Cálculo: Considere que existem M posições a serem preenchidas por N valores utilizando uma função aleatória *Hash*. O cálculo de **não** haver colisão pode ser descrito como segue:

$$p(M, N) = \left(\frac{M}{M}\right) \left(\frac{M-1}{M}\right) \left(\frac{M-2}{M}\right) \cdots \left(\frac{M-N+1}{M}\right)$$

$$p(M, N) = \prod_{k=1}^N \left(\frac{M-k+1}{M}\right)$$

$$p(M, N) = \frac{M!}{(M-N)!M^N}$$

Exemplo de Colisão

Alguns resultados para esse colisão:

Exemplo de Colisão

Alguns resultados para esse colisão:

Time de Futebol:

$$p(365, 23) = \frac{M!}{(M - N)!M^N} = \frac{365!}{(365 - 23)!365^{23}} \approx 49.27\%$$

Exemplo de Colisão

Alguns resultados para esse colisão:

Time de Futebol:

$$p(365, 23) = \frac{M!}{(M - N)!M^N} = \frac{365!}{(365 - 23)!365^{23}} \approx 49.27\%$$

Nossa turma (39 alunos):

$$p(365, 23) = \frac{365!}{(365 - 39)!365^{39}} \approx 12.17\%$$

Exemplo de Colisão

Alguns resultados para esse colisão:

Time de Futebol:

$$p(365, 23) = \frac{M!}{(M - N)!M^N} = \frac{365!}{(365 - 23)!365^{23}} \approx 49.27\%$$

Nossa turma (39 alunos):

$$p(365, 23) = \frac{365!}{(365 - 39)!365^{39}} \approx 12.17\%$$

Ou seja, apesar da nossa turma ter apenas 39 alunos, a chance de ter ao menos dois alunos completando idade no mesmo dia é de 87.82%.

Tratamento de colisões

Dois fatores reduzem a quantidade de colisões numa tabela Hash. São elas:

- Escolha do tamanho da tabela *Hash*.
- Escolha adequada da função *Hash*.

Tratamento de colisões

Dois fatores reduzem a quantidade de colisões numa tabela Hash. São elas:

- Escolha do tamanho da tabela *Hash*.
- Escolha adequada da função *Hash*.

Mas e quando nada mais pode ser feito?

Tratamento de colisões

Dois fatores reduzem a quantidade de colisões numa tabela Hash. São elas:

- Escolha do tamanho da tabela *Hash*.
- Escolha adequada da função *Hash*.

Mas e quando nada mais pode ser feito? Uso de **Tratamento de colisões**:

- Endereçamento Aberto:
- Endereçamento Encadeado:

Tratamento de Colisão

Endereçamento aberto

Endereçamento aberto

Caso ocorra uma colisão, percorre a tabela *Hash* em busca de um espaço vazio. Os elementos são armazenados diretamente na tabela (sem listas encadeadas).

Tipos de Endereçamento Aberto:

- Sondagem Linear
- Sondagem quadrática
- Duplo Hash

Endereçamento aberto

Sondagem Linear

Algoritmo:

Endereçamento aberto

Sondagem Linear

Algoritmo:

1. Posição da chave c é calculada por uma função *hash*.

$$p_x = p_c = \text{hash}(c)$$

Algoritmo:

1. Posição da chave c é calculada por uma função *hash*.

$$p_x = p_c = \text{hash}(c)$$

2. Chave é armazenada na posição p_x se estiver livre e Fim do algoritmo.

Algoritmo:

1. Posição da chave c é calculada por uma função *hash*.

$$p_x = p_c = \text{hash}(c)$$

2. Chave é armazenada na posição p_x se estiver livre e Fim do algoritmo.
3. $p_x \leftarrow p_x + 1$ e volta para o passo 2.

Características:

- Estratégia Simples;
- Provoca o **Agrupamento Primário**:
 - A medida que a tabela hash fica cheia, o tempo para incluir ou buscar um elemento aumenta;
 - A medida que os elementos são inseridos surgem longas sequências de posições ocupadas;
 - A ocorrência desses agrupamentos aumenta o **tempo de pesquisa**, diminuindo o desempenho.

Endereçamento aberto

Sondagem Quadrática

Tenta espalhar as chaves utilizando uma função do segundo grau. Utiliza quase o mesmo algoritmo da Sondagem Linear com a diferença que a sondagem é realizada pela fórmula:

$$p_x \leftarrow p_x + c_1(i - 1) + c_2(i - 1)^2$$

Onde i representa a tentativa de inserir a chave na posição p_c . Quando não há colisão, $p_x = p_c$.

$$p_x \leftarrow p_x + c_1(i - 1) + c_2(i - 1)^2$$

Sendo c_1 e c_2 coeficientes quaisquer escolhidos de forma a varrer a tabela *Hash*.

Características:

- Resolve o problema do agrupamento primária;
- Cria o problema do **agrupamento secundário**.
 - As chaves com mesmo p_c também terão os mesmos p_x .
 - A degradação do secundário é menor do que o primário.

Tenta espalhar os elementos utilizando duas funções *Hash*.

A atualização quando ocorre a colisão pode ser feita como segue:

$$px \leftarrow H_1 + H_2 * (i - 1)$$

- com $i > 1$ representando as tentativas em armazenar o elemento.
- Com H_1 e H_2 sendo duas funções hash aplicada a chave c .
- Obrigatoriamente essas funções devem ser diferentes e H_2 não pode retornar o valor de 0.

Tratamento de Colisão

Endereçamento Encadeado

- O endereço base pode ser encontrado para chaves diferentes como resultado da função de dispersão.
- Uma ideia para tratar colisões consiste em armazenar chaves sinônimas em listas encadeadas:
 - Encadeamento Exterior
 - Encadeamento Interior

Endereçamento Encadeado

Encademento Exterior

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

Endereçamento Encadeado

Encademento Exterior

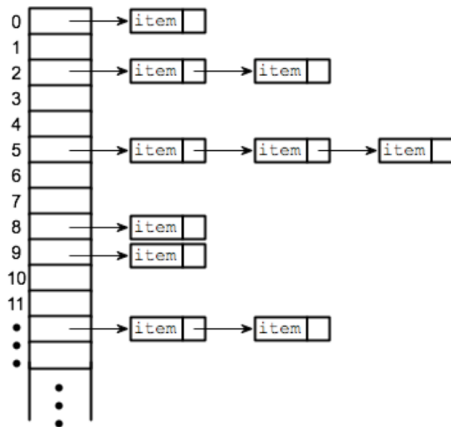


Figura: Encadeamento Exterior

Endereçamento Encadeado

Encadramento Interior

Endereçamento Encadeado

Encadramento Interior

- Não queremos adicionar m listas externas à tabela

Endereçamento Encadeado

Encadramento 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.

Endereçamento Encadeado

Encadramento 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.
- 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 .

Endereçamento Encadeado

Encadramento 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.
- 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 a tabela tem tamanho $m = p + s$.

Endereçamento Encadeado

Encademento 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.
- 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 a tabela tem tamanho $m = p + s$.
 - Os valores de p e s são fixos.

Endereçamento Encadeado

Encadramento 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.
- 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 a tabela tem tamanho $m = p + s$.
 - Os valores de p e s são fixos.
- A função de dispersão deve obter endereços-base na faixa $[0, p - 1]$ apenas.

Endereçamento Encadeado

Encademento Interior

Exemplo:

Endereçamento Encadeado

Encadramento Interior

Exemplo:

- Número de chaves: $N = 5$; (48 03 80 31 20);

Endereçamento Encadeado

Encadramento Interior

Exemplo:

- Número de chaves: $N = 5$; (48 03 80 31 20);
- Tamanho da tabela *Hash*: $m = 7$

Endereçamento Encadeado

Encademento Interior

Exemplo:

- Número de chaves: $N = 5$; (48 03 80 31 20);
- Tamanho da tabela *Hash*: $m = 7$
- Destinos endereçáveis pela função *Hash*: $p = 4$.

Endereçamento Encadeado

Encademento Interior

Exemplo:

- Número de chaves: $N = 5$; (48 03 80 31 20);
- Tamanho da tabela *Hash*: $m = 7$
- Destinos endereçáveis pela função *Hash*: $p = 4$.
- Expansão: $s = 3$

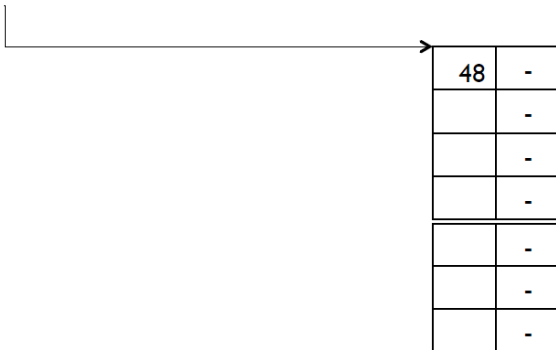
Endereçamento Encadeado

Encademento Interior

$$m = 7$$

$$48 \bmod 4 = 0$$

48 03 80 31 20

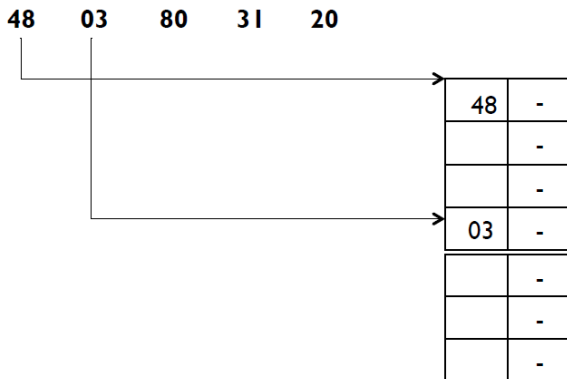


Endereçamento Encadeado

Encademento Interior

$$m = 7$$

$$03 \bmod 4 = 3$$

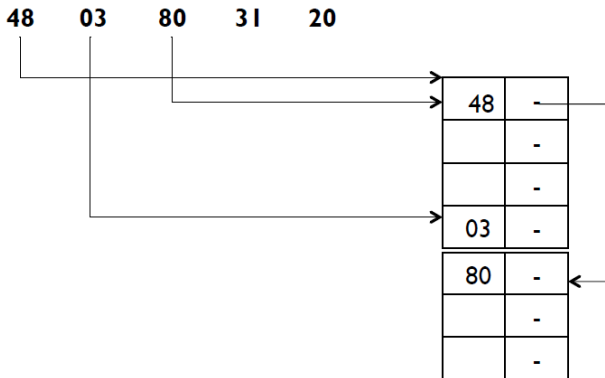


Endereçamento Encadeado

Encademento Interior

$$m = 7$$

$$80 \bmod 4 = 0$$

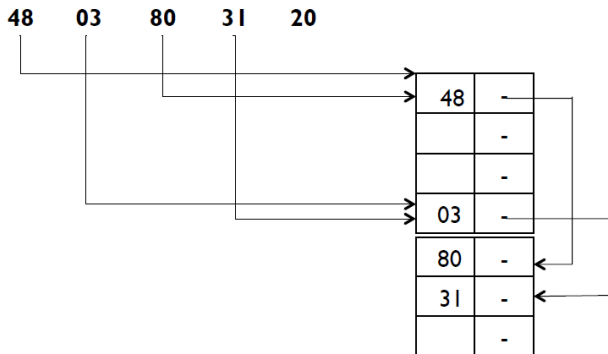


Endereçamento Encadeado

Encademanto Interior

$$m = 7$$

$$31 \bmod 4 = 3$$



Endereçamento Encadeado

Encademento Interior

$$m = 7$$

$$20 \bmod 4 = 0$$

