UFRRJ

# Estrutura de Dados 2

## Tabelas de Dispersão
### Tratamento de Colisões

UFRRJ

# Introdução

- Fator de carga de uma tabela hash é **α = n/m**, onde n é o número de registros na tabela
  - O número de colisões aumenta conforme o fator de carga aumenta
  - Uma forma de diminuir o número de colisões é diminuir o fator de carga
- Como tratar as colisões?

# Tratamento de colisões

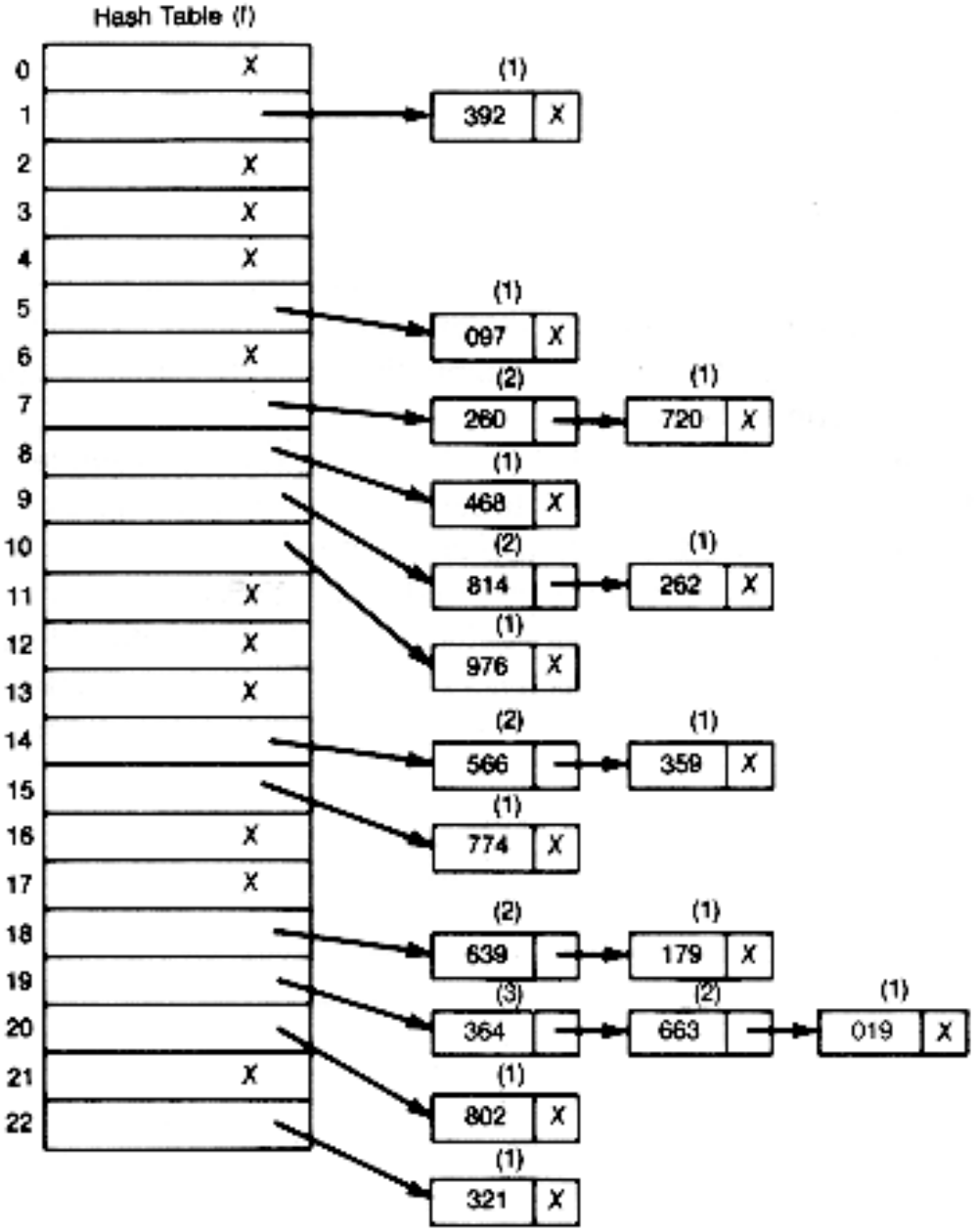
- Por encadeamento
  - Encadeamento exterior
  - Encadeamento interior
- Por endereçamento aberto

# Encadeamento Exterior

- Manter *m* listas encadeadas, uma para cada posição da tabela
  - A tabela em si não possui nenhum registro, apenas ponteiros para as listas encadeadas

**Por isso se chama encadeamento *exterior***

# Encadeamento Exterior



# Encadeamento Exterior

- Busca por um registro de chave **x**:
  1. Calcular o endereço usando **h(x)**
  2. Percorrer a lista encadeada associada ao endereço, procurando pela chave desejada
  3. Caso a chave não seja encontrada na lista encadeada, ela não existe na tabela.

# Encadeamento Exterior

- E como seria a inserção?

# Encadeamento Exterior

- Inserção de um registro de chave **x**:
  1. Calcular o endereço usando **h(x)**
  2. Buscar o registro na lista encadeada do endereço calculado
  3. Se o registro for encontrado, sinalizar erro
  4. Caso contrário, inserir o registro ao final da lista encadeada

# Encadeamento Exterior

- Exclusão de um registro de chave **x**:
  1. Calcular o endereço usando **h(x)**
  2. Buscar o registro na lista encadeada do endereço calculado
  3. Se o registro for encontrado, exclui-lo da lista
  4. Caso contrário, sinalizar erro

# Encadeamento Exterior

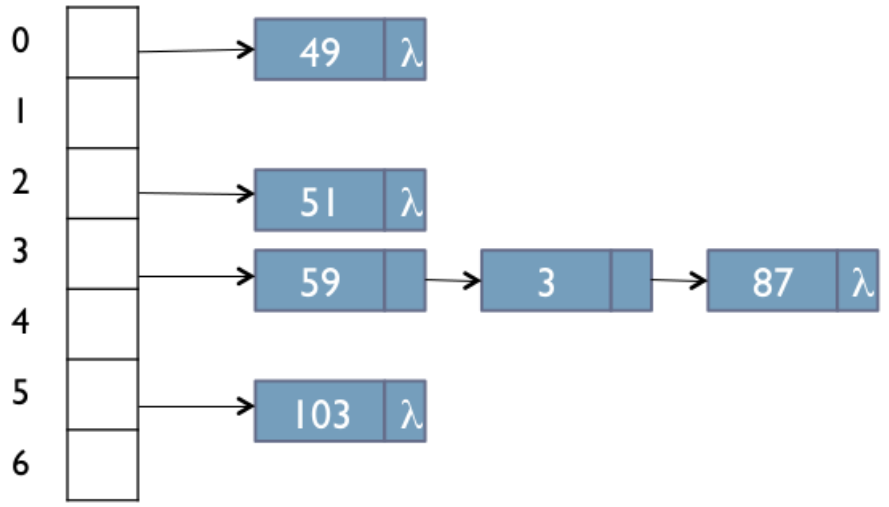
- Como seria a complexidade no pior caso?
  - Necessário percorrer a lista até o final para concluir que a chave não se encontra lá
  - Comprimento da lista encadeada pode ser O(n)
  - Complexidade no pior caso: O(n)

# Encadeamento Exterior

- Como seria a complexidade caso médio?
  - Assumindo que **h(x)** é uniforme, todos os compartimentos (buckets) estão preenchidos com o mesmo número de registros
  - Busca sem sucesso: n/m
  - Busca com sucesso: n/m
  - Se n = m, a complexidade se torna O(1), ou seja, constante

# Encadeamento Exterior

- Implementação




Arquivo

| | | |
|---|---|---|
| 0 | 7 | |
| 1 | -1 | |
| 2 | 11 | |
| 3 | 8 | |
| 4 | -1 | |
| 5 | 9 | |
| 6 | -1 | |
| 7 | 49 | -1 |
| 8 | 59 | 10 |
| 9 | 103 | -1 |
| 10 | 3 | 12 |
| 11 | 51 | -1 |
| 12 | 87 | -1 |
| 13 | | |

m = 7

# Encadeamento Exterior

- Utilização do flag de **status** para cada registro
  - Quando o compartimento tem um registro, marcar o flag como OCUPADO
  - Quando o compartimento está vazio, marcar o flag como LIVRE

# Encadeamento Exterior

- Como seriam os procedimentos para inclusão e exclusão?

# Encadeamento Exterior

- Exclusão:
  - Marcar o flag de status do registro como LIVRE
  - É necessário acertar os ponteiros?

# Encadeamento Exterior

- Inclusão:
  - Adicionar o novo registro sempre ao final da lista encadeada

- Periodicamente deve-se rearrumar o arquivo para ocupar as posições onde o flag de status está LIVRE

# Encadeamento Exterior

- Inclusão (opçao 2):
  - Ao procurar pelo registro na lista encadeada, guardar o endereço *p* do primeiro registro encontrado com status LIVRE
  - Se chegar ao final e a chave não for encontrada, armazená-la em *p*
  - Atualizar os ponteiros:
    - Nó anterior deve apontar para o registro inserido
    - Nó inserido deve apontar para o nó apontado pelo anterior

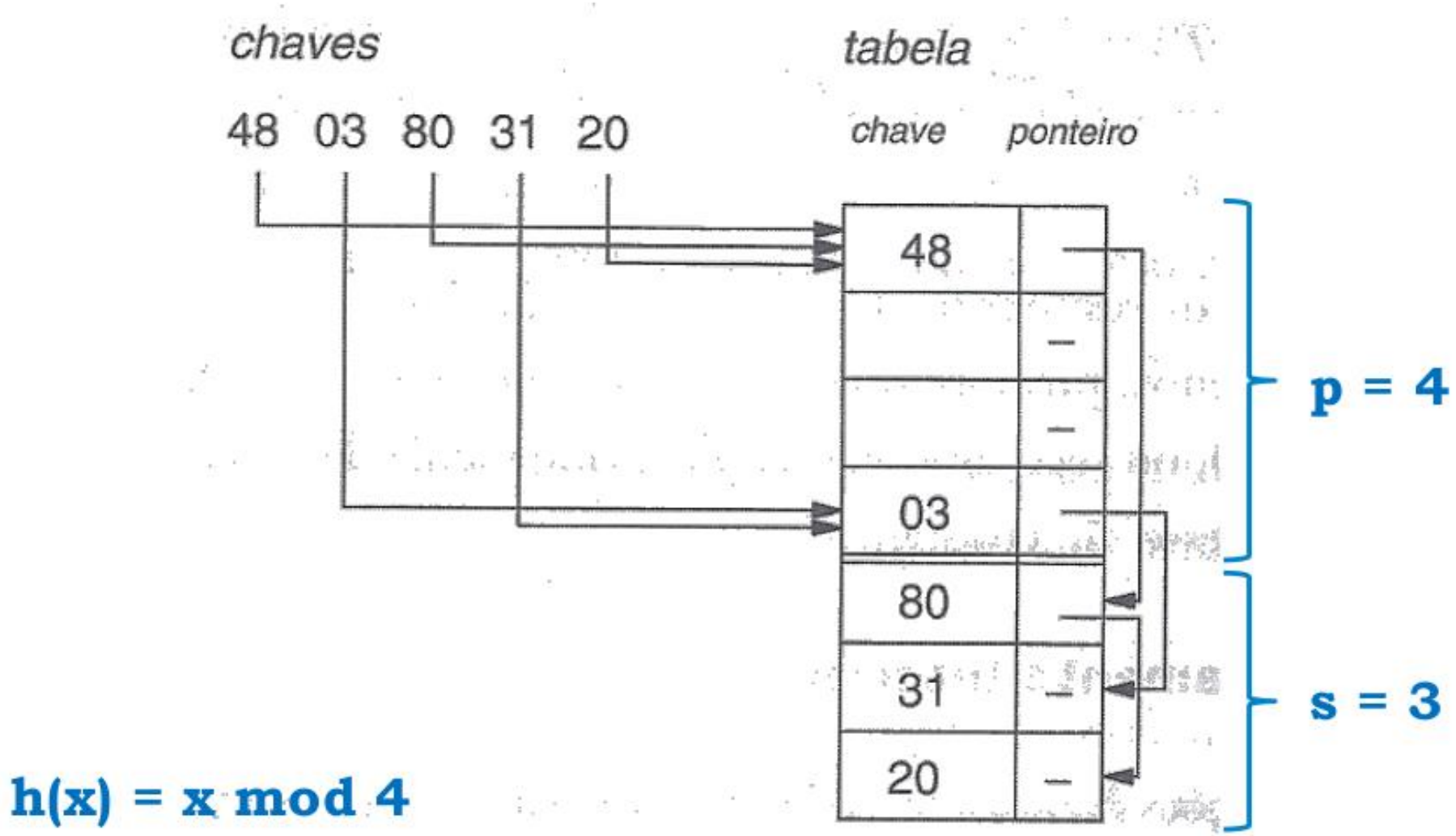
# Encadeamento Interior

- Mesmo nas situações onde não é possível permitir que o número de registros cresça indefinidamente, é possível tratar colisões
- Para tanto, pode-se utilizar duas técnicas:
  - Com zona de colisões
  - Sem zona de colisões

# Encadeamento Interior

- Com Zona de Colisões:
- Divide-se a tabela em duas zonas:
  - Uma de endereços-base, com tamanho *p*
  - Uma para tratar colisões, com tamanho *s*
  - $m = s + p$
  - h(x) deve gerar endereços entre [0, p-1]
- Overflow ocorre quando tenta-se inserir um registro em uma tabela cheia

# Encadeamento Interior



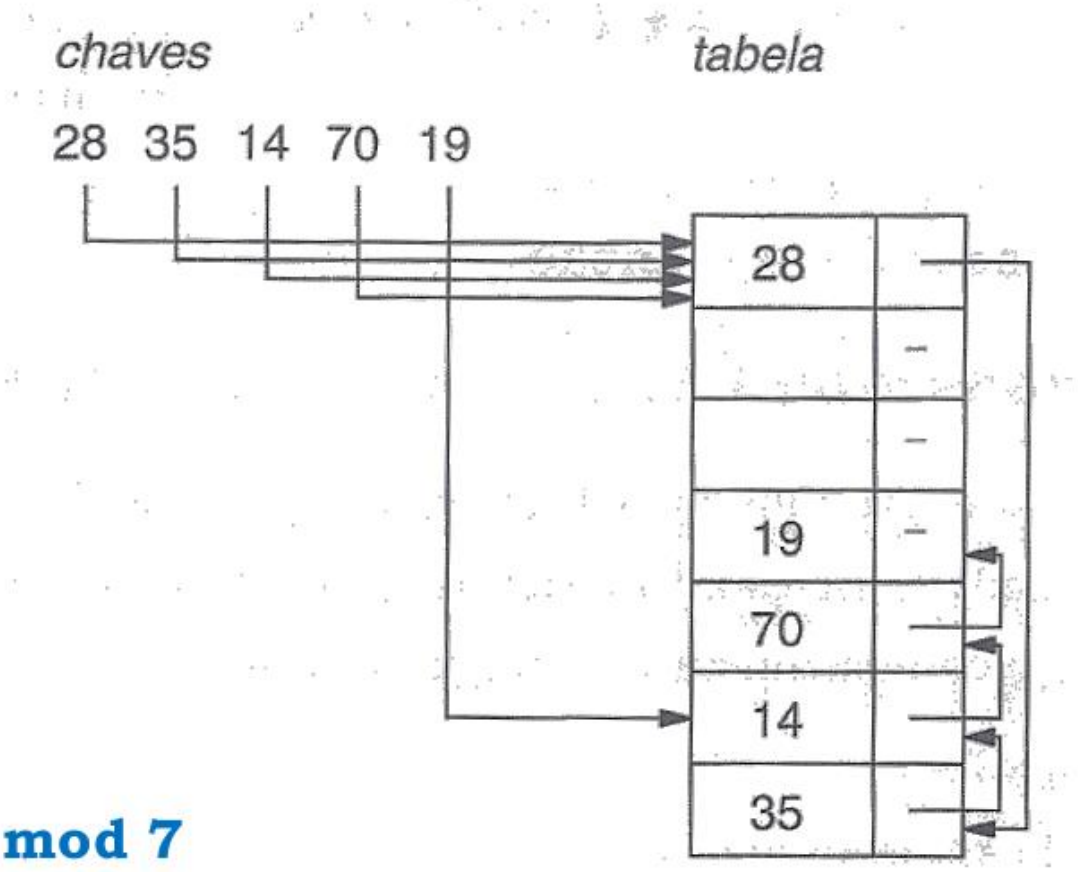
# Reflexões

- O que aconteceria se
  - *p* fosse muito grande e *s* muito pequeno ?
  - *s* fosse muito grande e *p* muito pequeno ?

# Encadeamento Interior

- Sem Zona de Colisões
  - Qualquer endereço da tabela pode ser de base ou de colisão
  - Quando há colisão, o registro é inserido no *primeiro compartimento vazio* a partir do compartimento que ocorreu a colisão
  - Efeito indesejado: colisões secundárias
    - Provenientes da coincidência de endereços para chaves que não são sinônimas

# Encadeamento Interior




**h(x) = x mod 7**

UFRRJ

# Exercício

1. Implementar em Java, na tabela de dispersão criada no exercício anterior, o tratamento para colisões utilizando Encadeamento Exterior
2. Escrever, em pseudo-código, os algoritmos para busca, inserção e remoção de chaves em uma tabela de dispersão de tamanho *m* e que utilize Encadeamento Interior Sem Zona de Colisões