ED62A-COM2A
ESTRUTURAS DE DADOS

Aula 08 - Tabelas Hash

Prof. Rafael G. Mantovani

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Apucarana - PR, Brasil

Universidade Tecnológica Federal do Paraná (UTFPR)
Engenharia de Computação

Roteiro

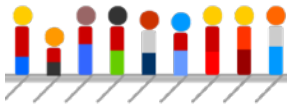
1. Introdução
2. Tabelas de Endereçamento Direto
3. Tabelas de Espalhamento
4. Resolução de colisões
5. Funções Hash
6. Referências

Roteiro

1. Introdução
2. Tabelas de Endereçamento Direto
3. Tabelas de Espalhamento
4. Resolução de colisões
5. Funções Hash
6. Referências

Introdução

Pilhas



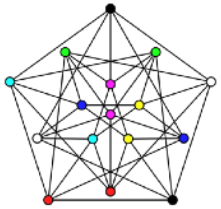
Filas

Árvores

Problema

Vetores

Grafos



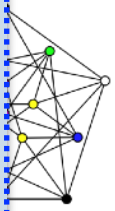
Etc

Introdução

Pilhas

Vetores

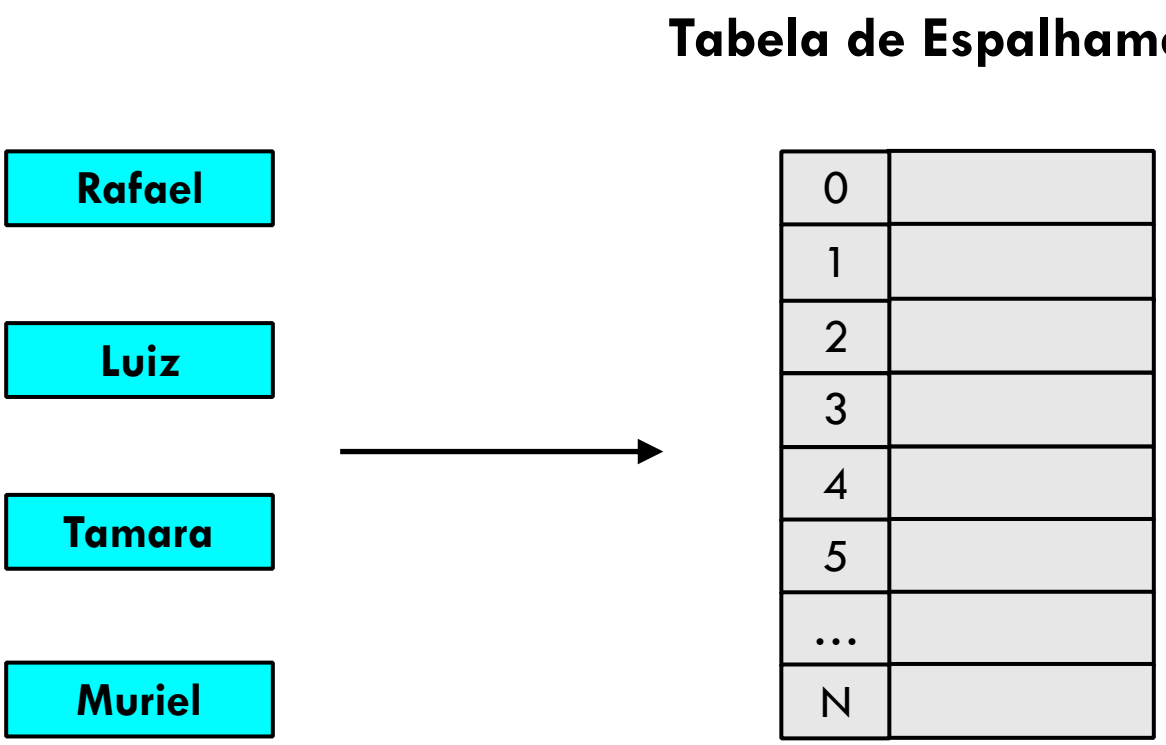
- Maioria das aplicações requer apenas 3 operações:
- Inserção, Remoção, Pesquisa
- Otimizá-las pode otimizar a solução codificada



Árvores

Etc

Introdução



Introdução

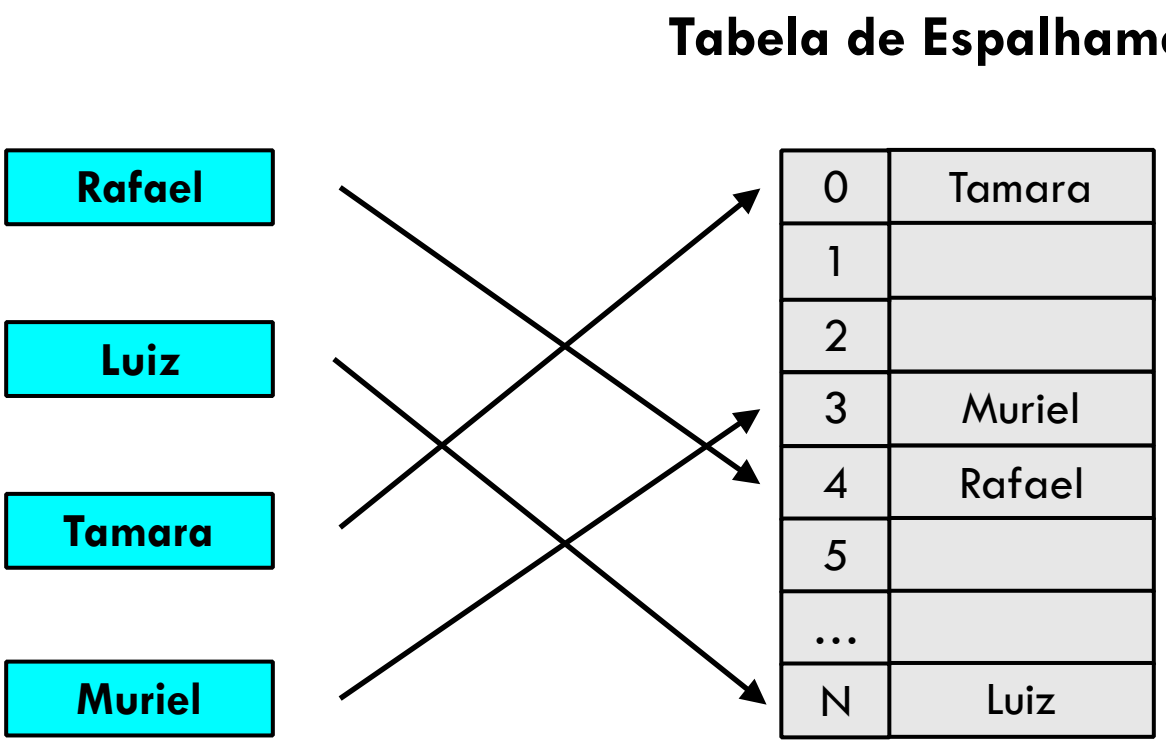
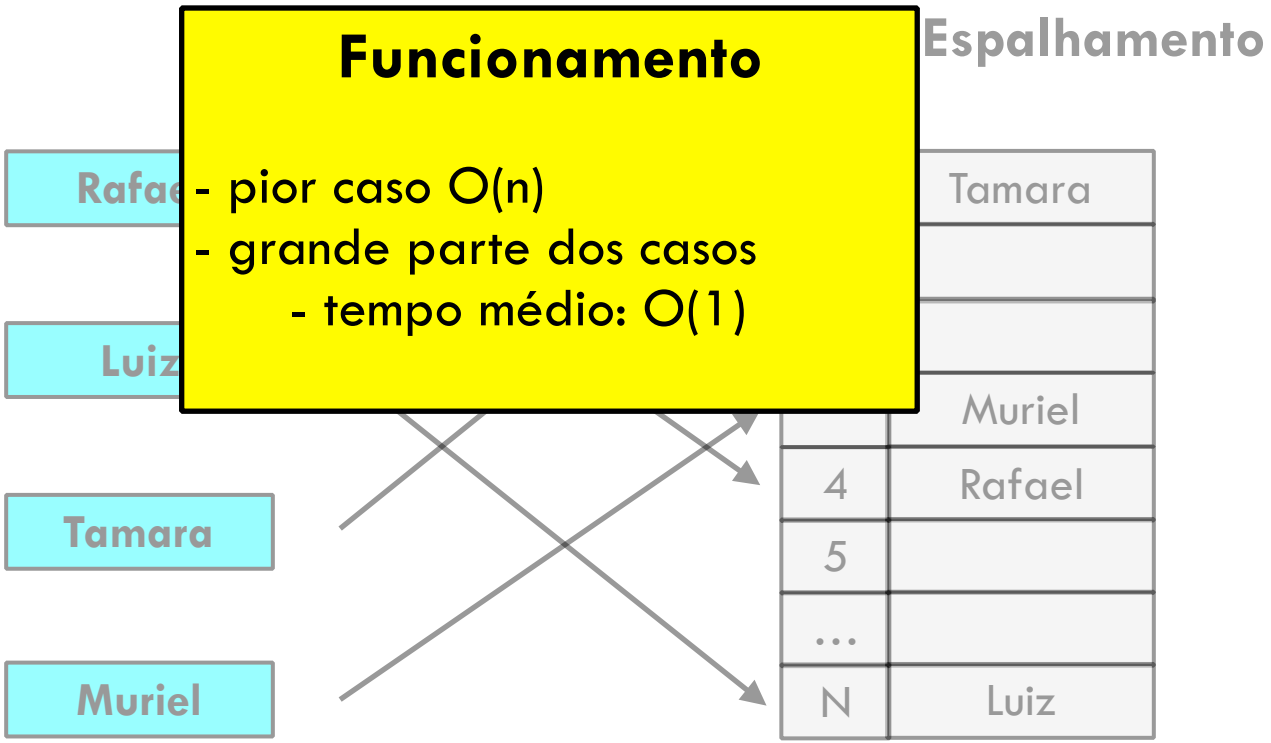
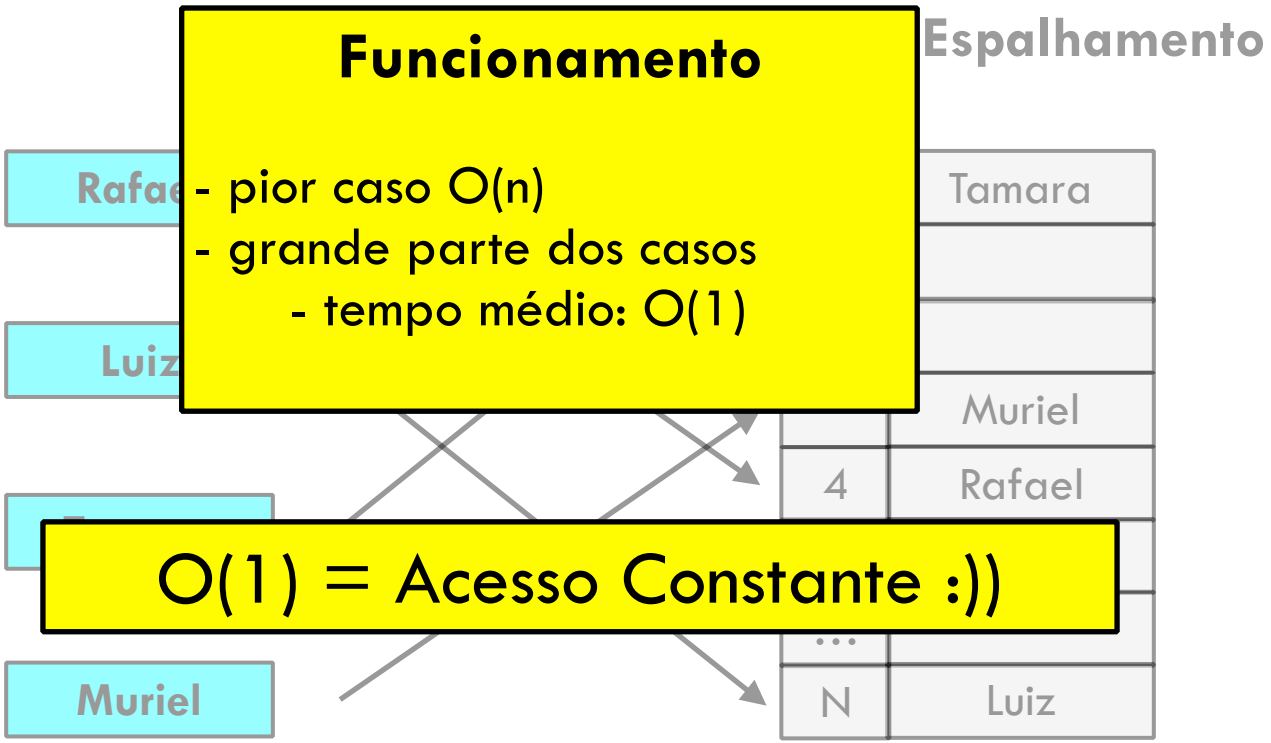


Tabela de Espalhamento
Rafael
Luiz
Tamara
Muriel
0
Tamara
1
2
3
Muriel
4
Rafael
5
...
N
Luiz

Introdução


Funcionamento
- pior caso O(n)
- grande parte dos casos
- tempo médio: O(1)
Espalhamento
Rafae
Luiz
Tamara
Muriel
Tamara
Muriel
4
Rafael
5
...
N
Luiz

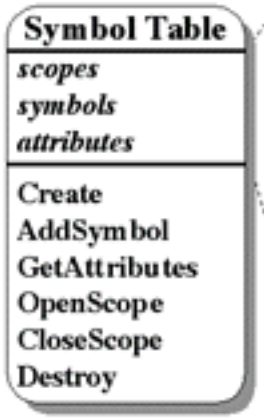
Introdução

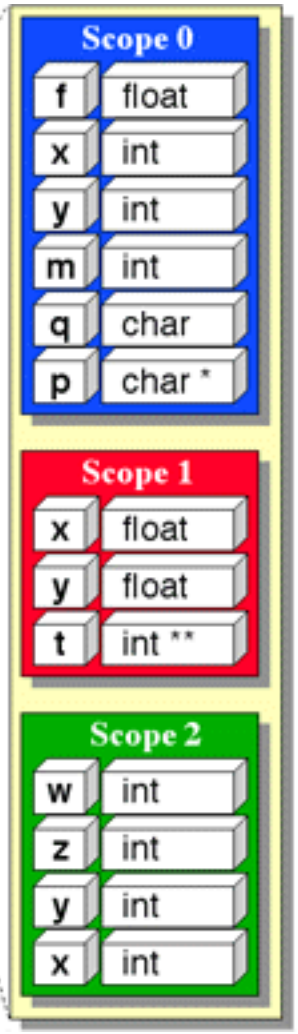

Funcionamento
- pior caso O(n)
- grande parte dos casos
- tempo médio: O(1)
Espalhamento
Rafae
Luiz
Muriel
Tamara
Muriel
4
Rafael
N
Luiz
O(1) = Acesso Constante :))

Introdução

Ex: Tabela(s) de Símbolos (compilador)

```
float    f;
int      x, y, m;
char     q, *p;
. . .
{
         float    x, y;
         int      **t;
         . . .
         {
                  int      w, z, y, x;
                  . . .
```





Introdução

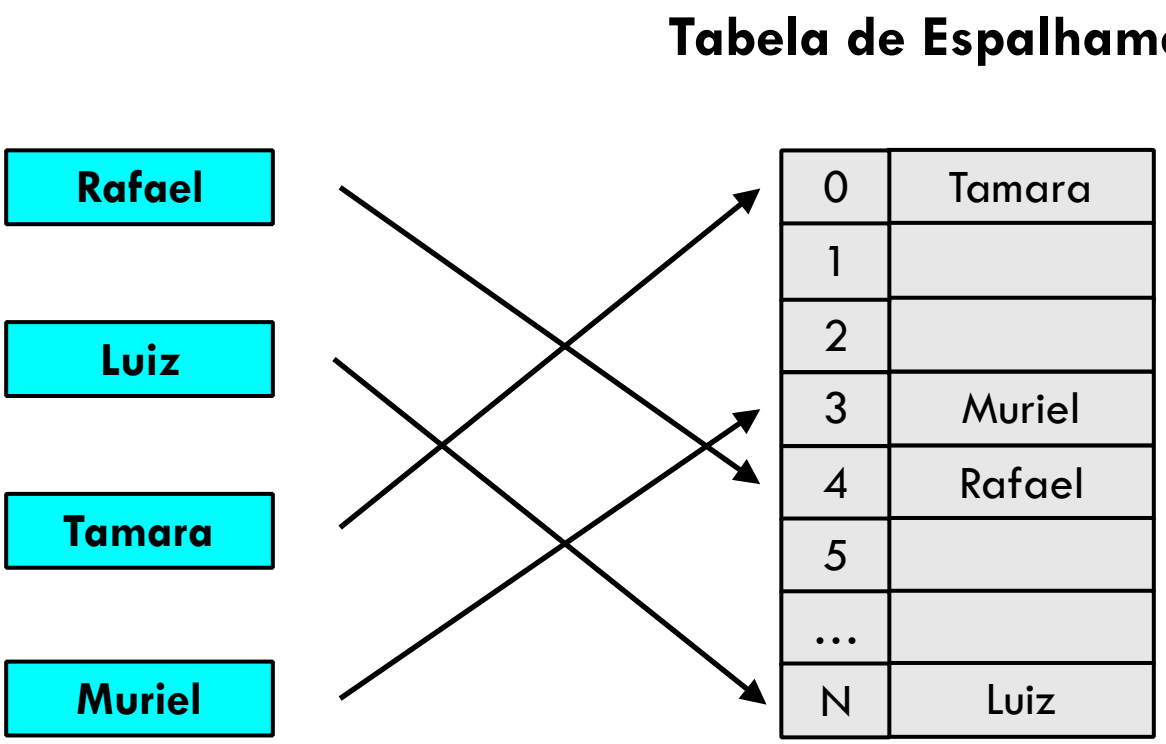
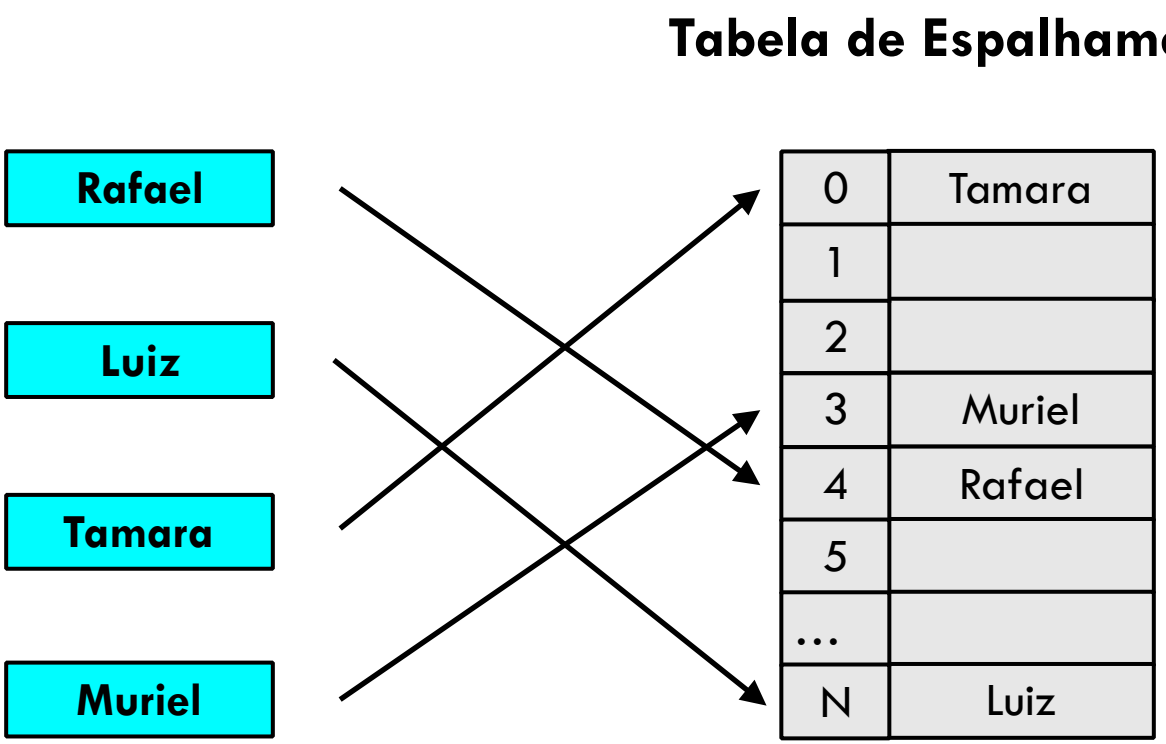


Tabela de Espalhamento
Rafael
Luiz
Tamara
Muriel
0
Tamara
1
2
3
Muriel
4
Rafael
5
...
N
Luiz

Introdução



Como é feito o espalhamento?

Roteiro

1. Introdução
2. **Tabelas de Endereçamento Direto**
3. Tabelas de Espalhamento
4. Resolução de colisões
5. Funções Hash
6. Referências

Tabela de Endereçamento Direto

Tabela

0	Tamara
1	Luiz
2	Rafael
3	Muriel
4	
5	

Tabela de Endereçamento Direto

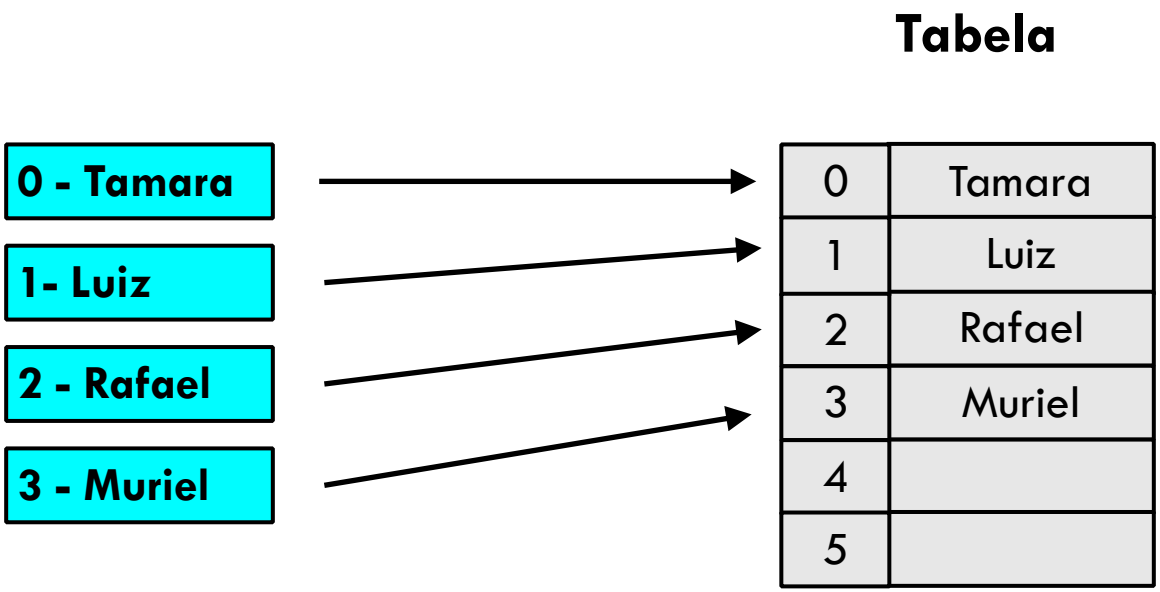
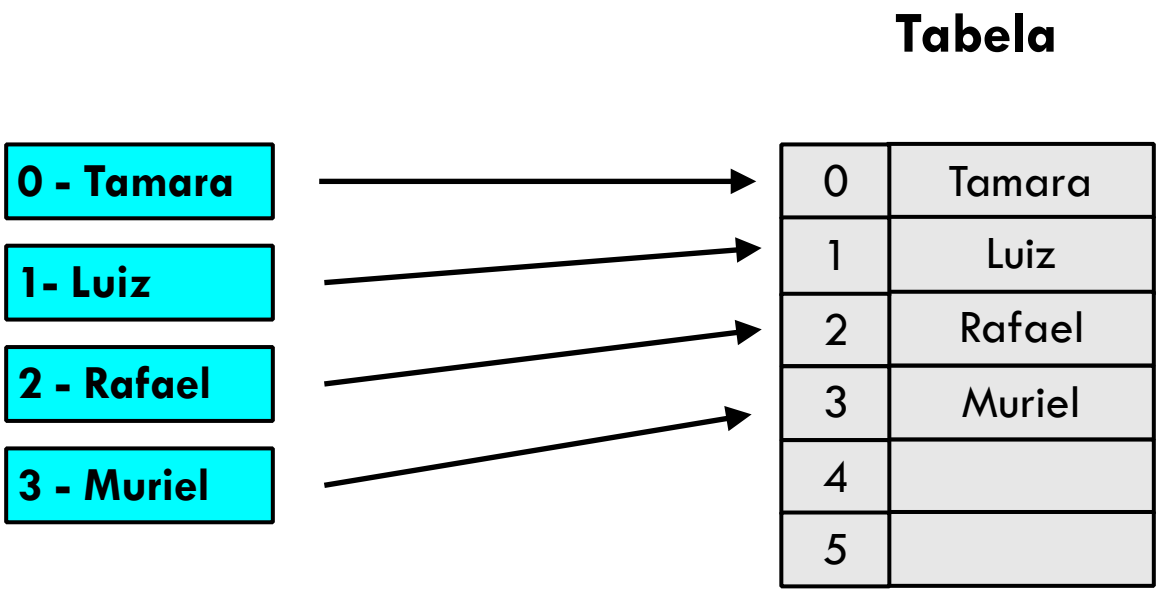


Tabela de Endereçamento Direto



Simples, funciona quando o universo de chaves (U) é razoavelmente pequeno.

Tabela de Endereçamento Direto

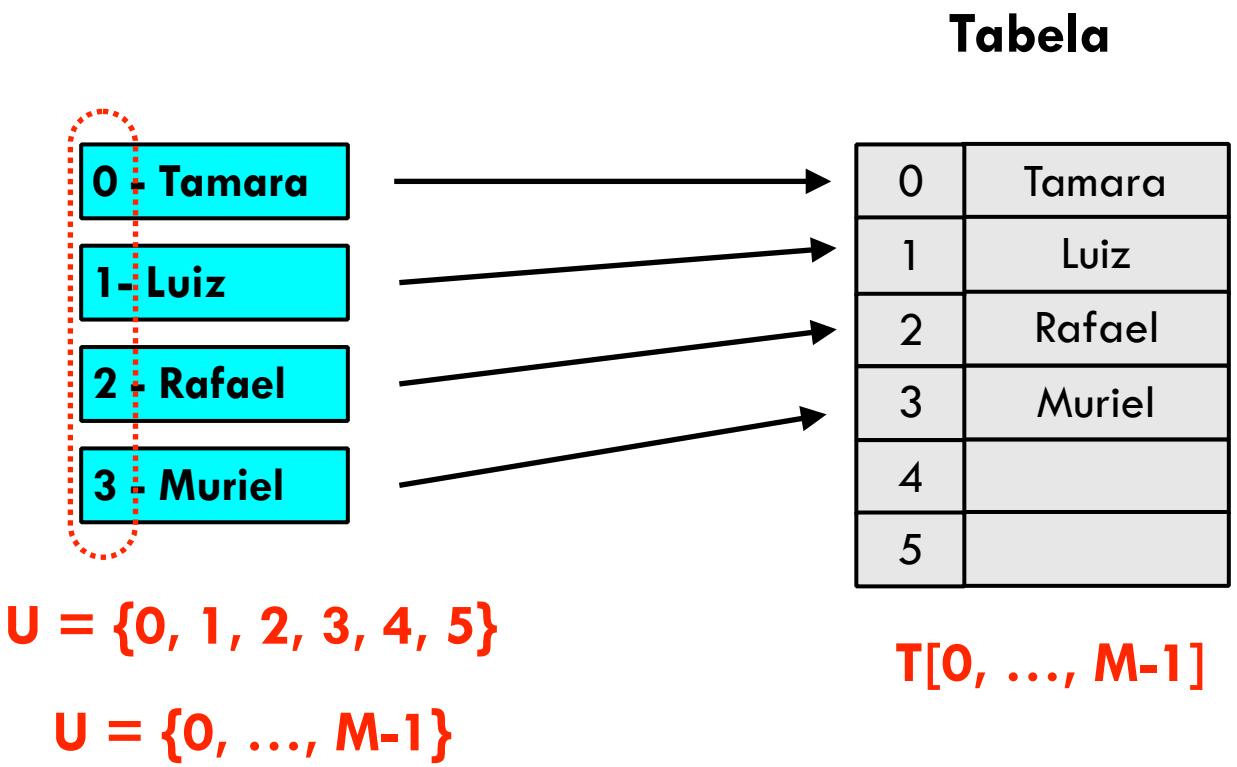
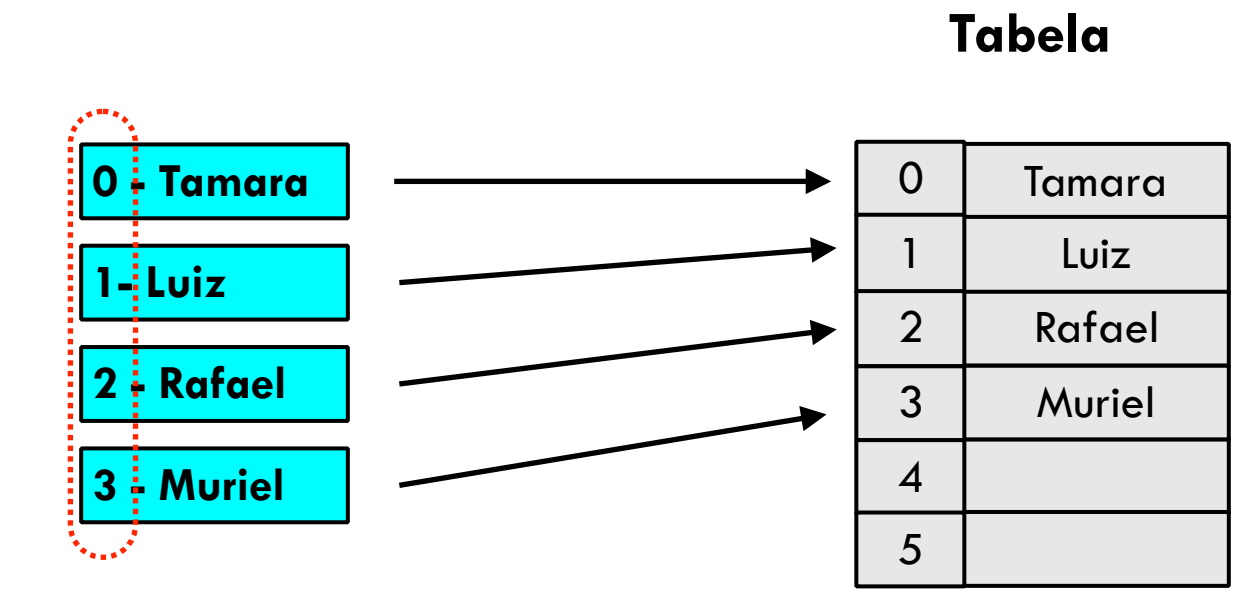
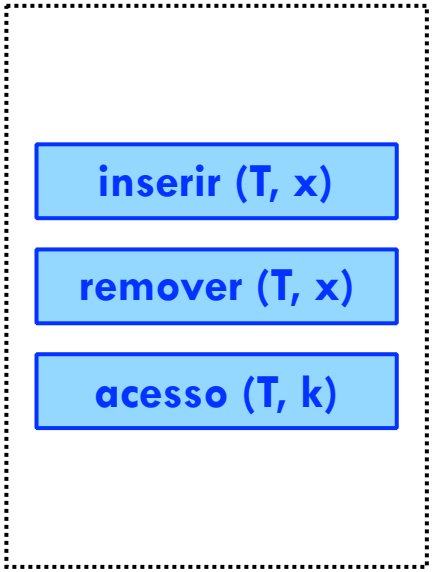


Tabela de Endereçamento Direto



Posição **K** aponta para o elemento com chave **K**.
Se o conjunto em **K** é vazio, então **T[K]** = NULL

Tabela de Endereçamento Direto



Operações de modificação

Tabela de Endereçamento Direto

- **Operações de Modificação**

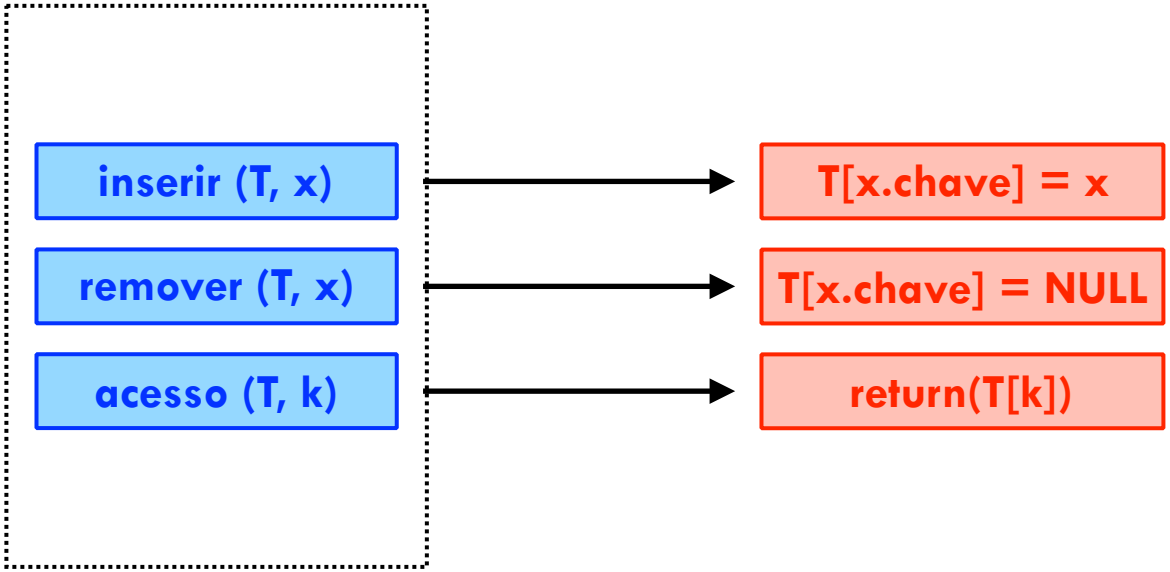
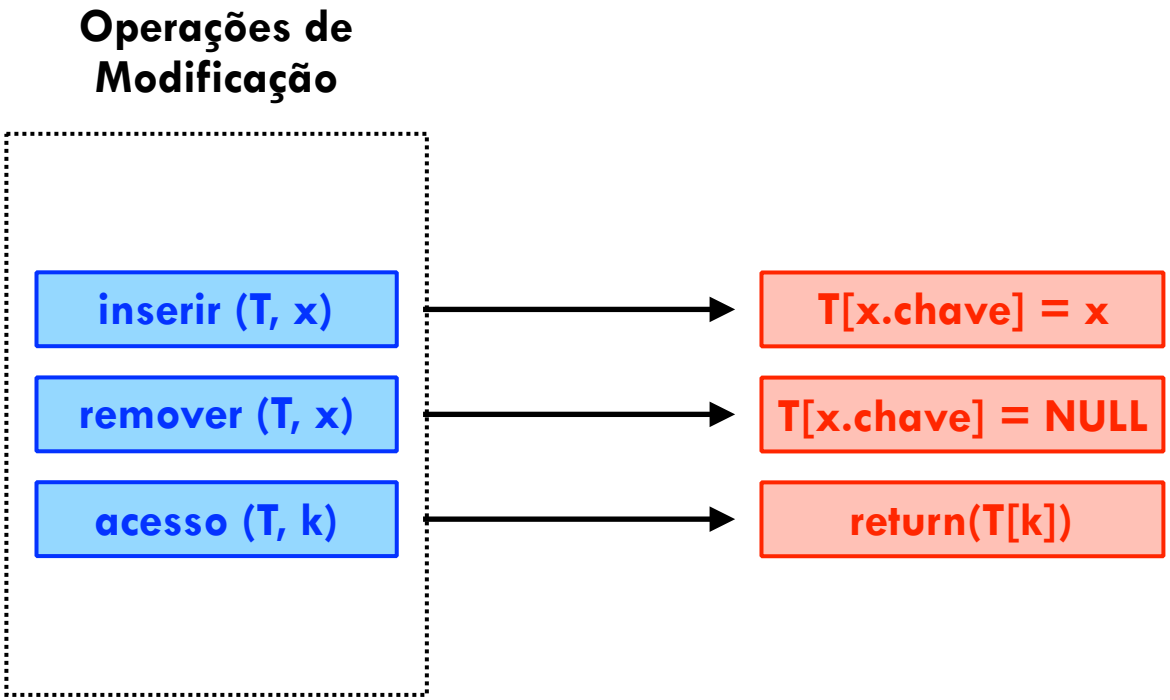


Tabela de Endereçamento Direto



tempo das operações >> O(1)

Exercício 01

- Dado um conjunto S de itens formado por uma tabela T[0 … M-1]. Faça uma função em C para computar o elemento máximo de S.
- Qual o desempenho no pior caso ?

Roteiro

1. Introdução
2. Tabelas de Endereçamento Direto
3. **Tabelas de Espalhamento**
4. Resolução de colisões
5. Funções Hash
6. Referências

Tabela de Espalhamento

0	Tamara
1	Luiz
2	Rafael
3	Muriel
4	
5	

Endereçamento Aberto

Problemas?

Tabela de Espalhamento

0	Tamara
1	Luiz
2	Rafael
3	Muriel
4	
5	

Endereçamento Aberto

Problemas?

- se U é grande, consome muita memória

Tabela de Espalhamento

0	Tamara
1	Luiz
2	Rafael
3	Muriel
4	
5	

Endereçamento Aberto

Problemas?

- se U é grande, consome muita memória

- chaves armazenadas < chaves totais (muitas posições nulas)

Tabela de Espalhamento

0	Tamara
1	Luiz
2	Rafael
3	Muriel
4	
5	

Endereçamento Aberto

Problemas?

- se U é grande, consome muita memória

- chaves armazenadas < chaves totais (muitas posições nulas)

Solução !

- reduzir o tamanho da tabela

- "Espalhar" as chaves

Tabela de Espalhamento

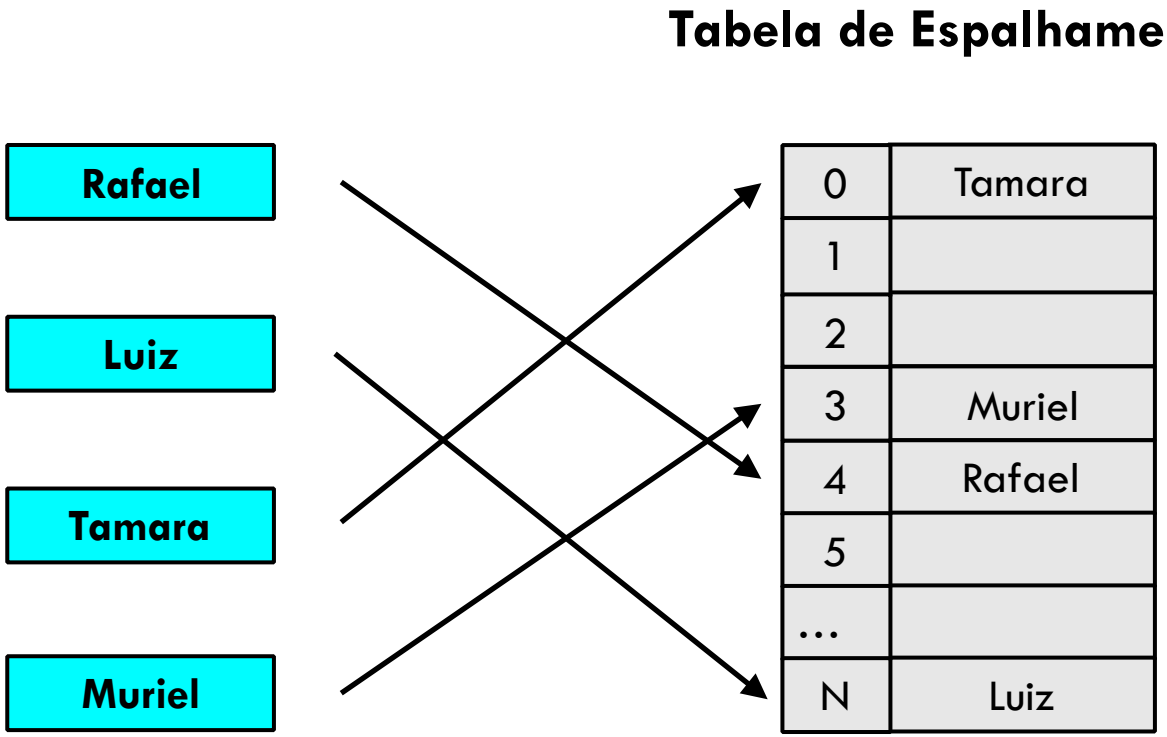
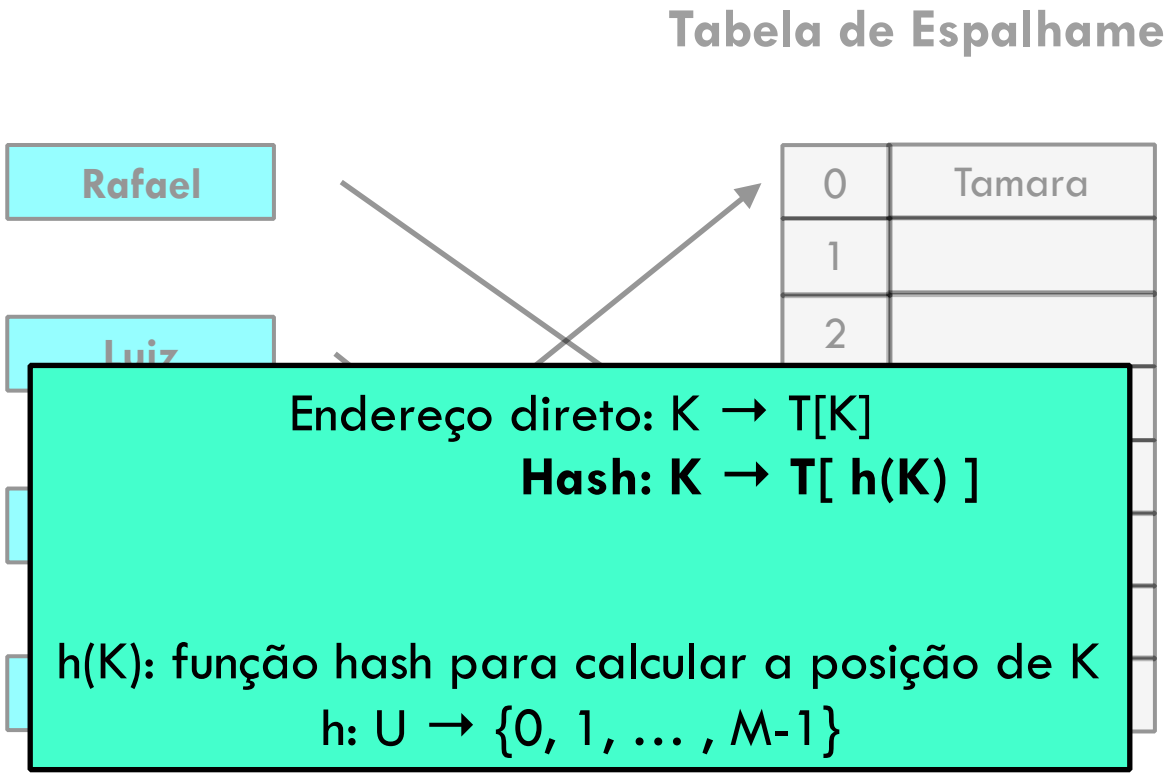


Tabela de Espalhamento



"Um elemento com a chave K **se espalha** até a posição h(K)"
"h(K) é o valor hash de K

Tabela de Espalhamento

- **Operações de Modificação**

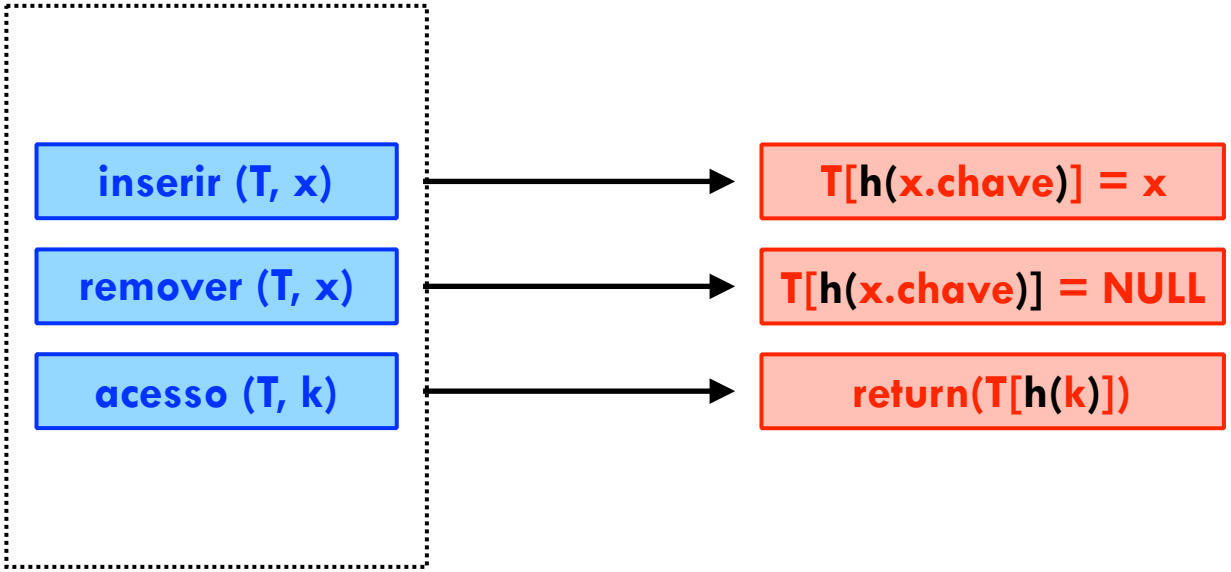
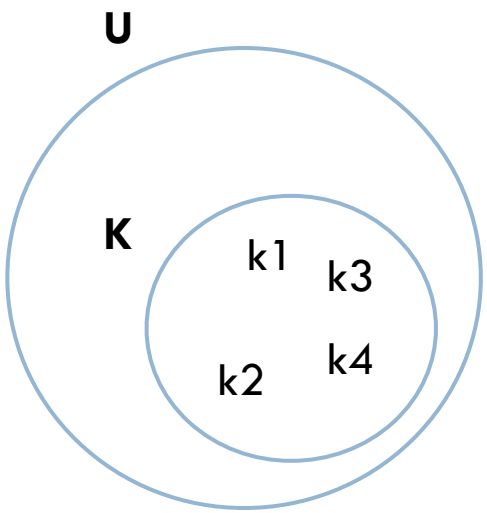


Tabela de Espalhamento



T

0	
1	
2	
3	
4	
5	
...	
N	

Tabela de Espalhamento

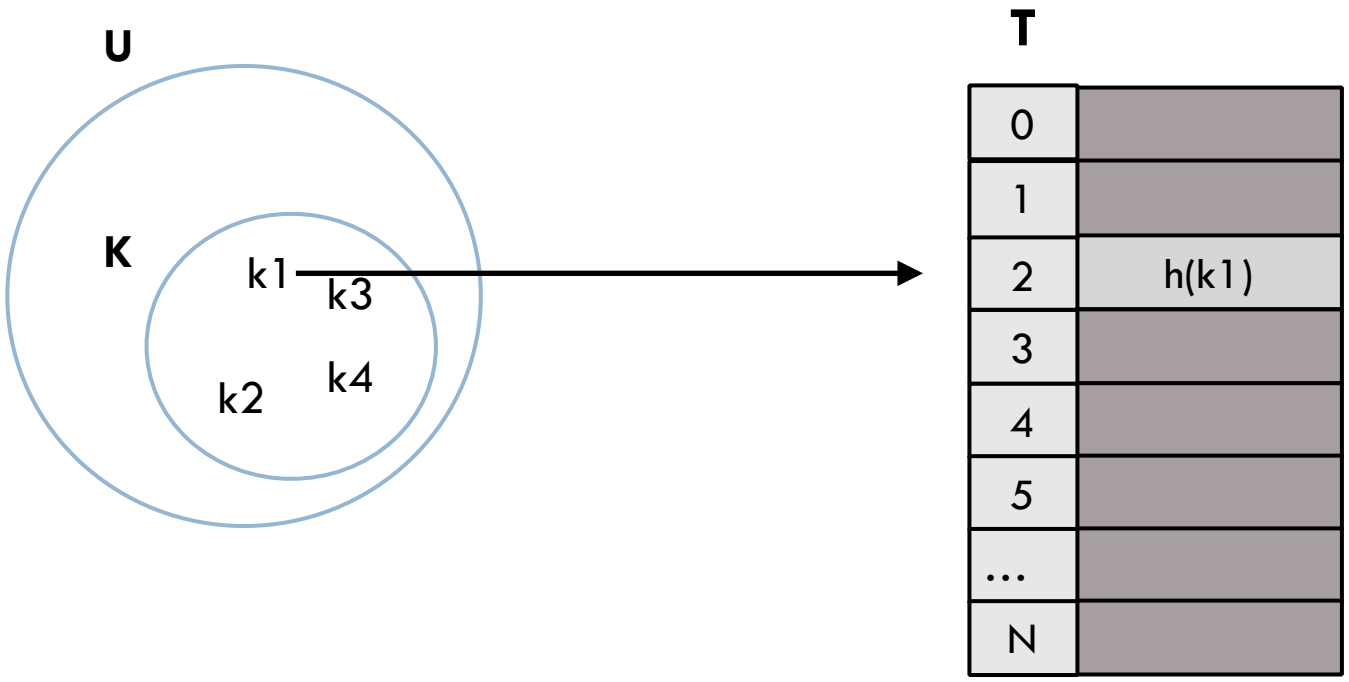


Tabela de Espalhamento

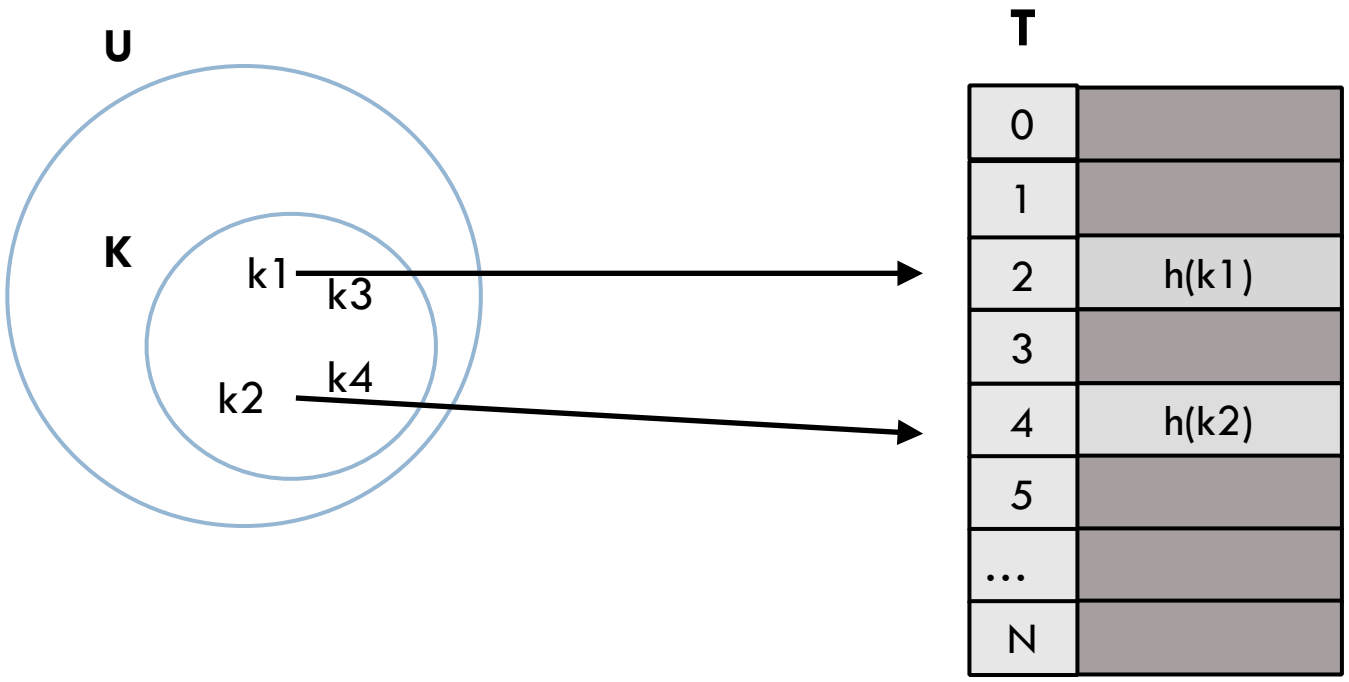


Tabela de Espalhamento

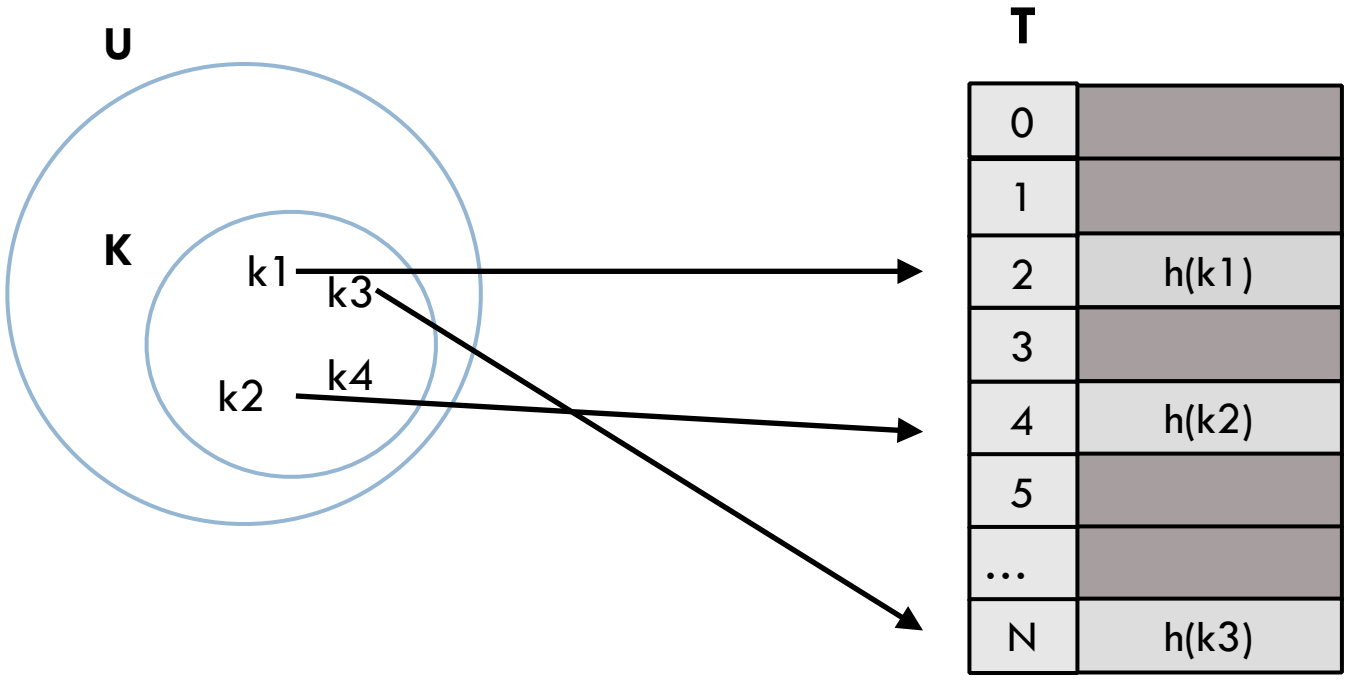


Tabela de Espalhamento

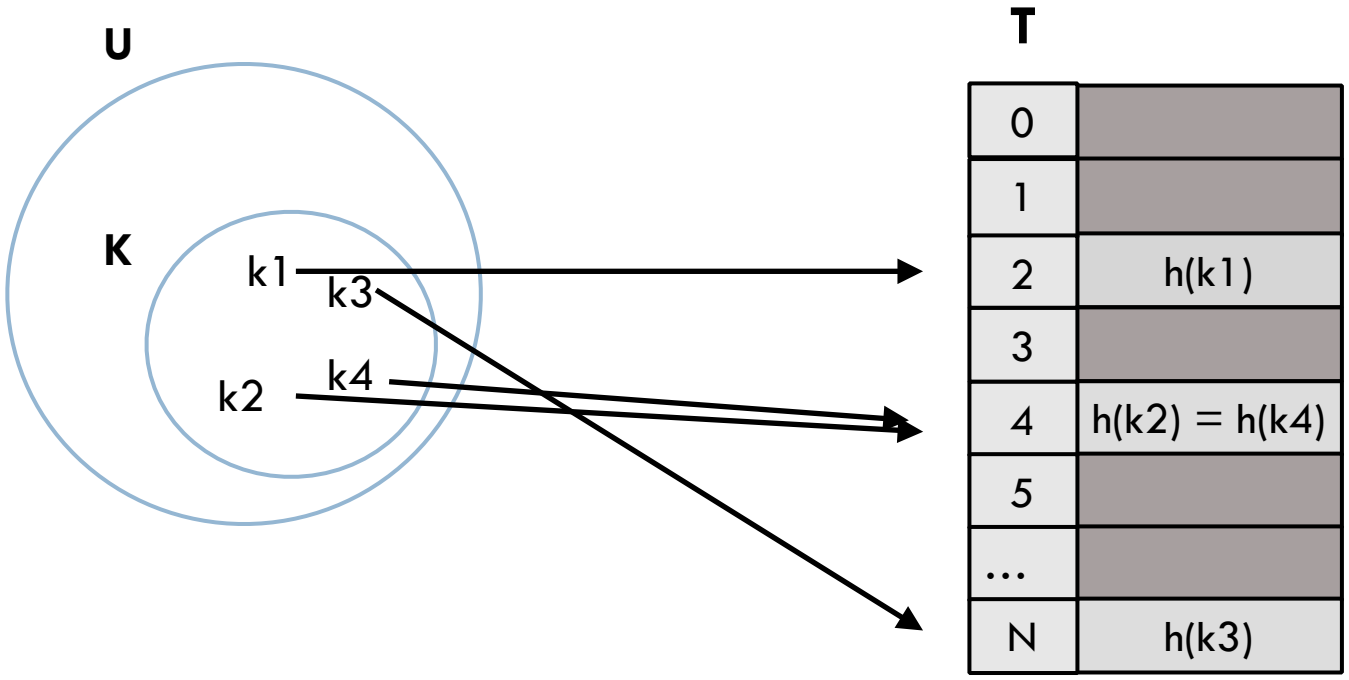
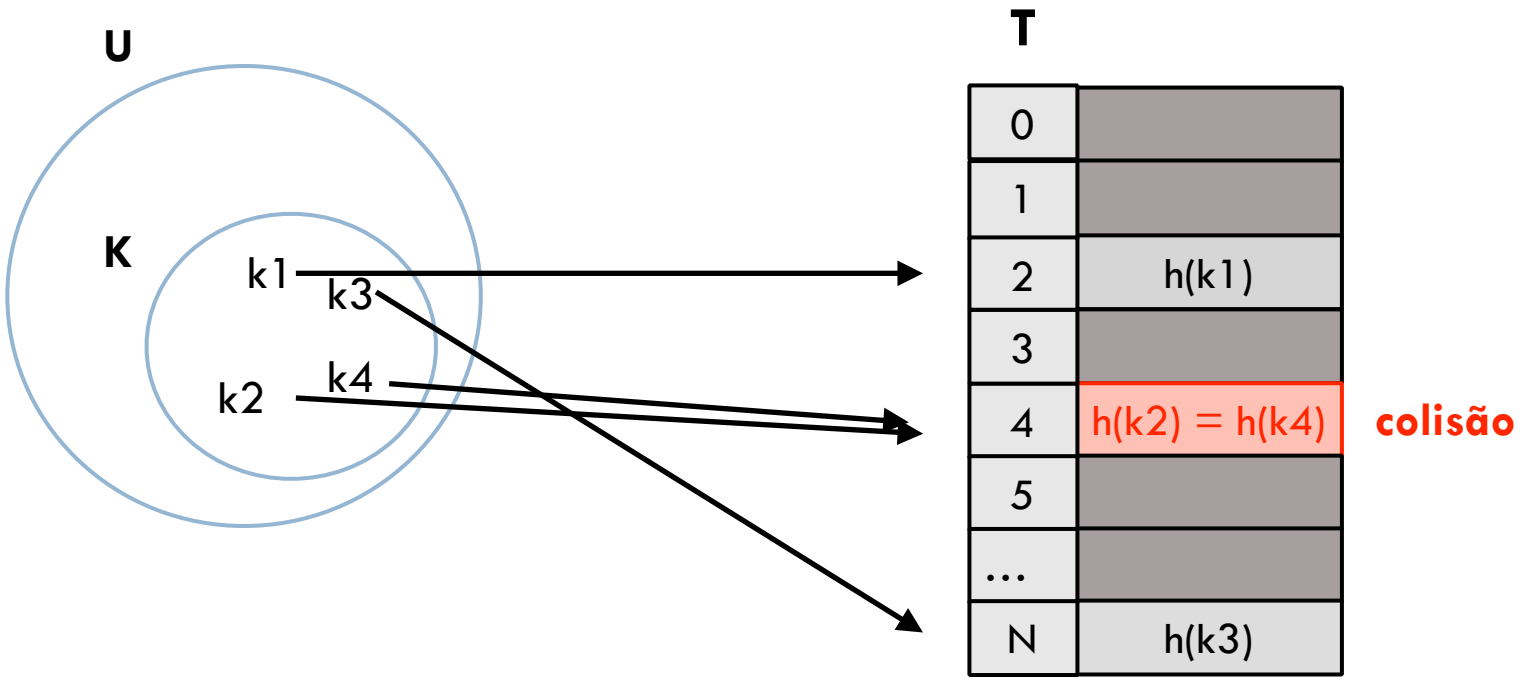
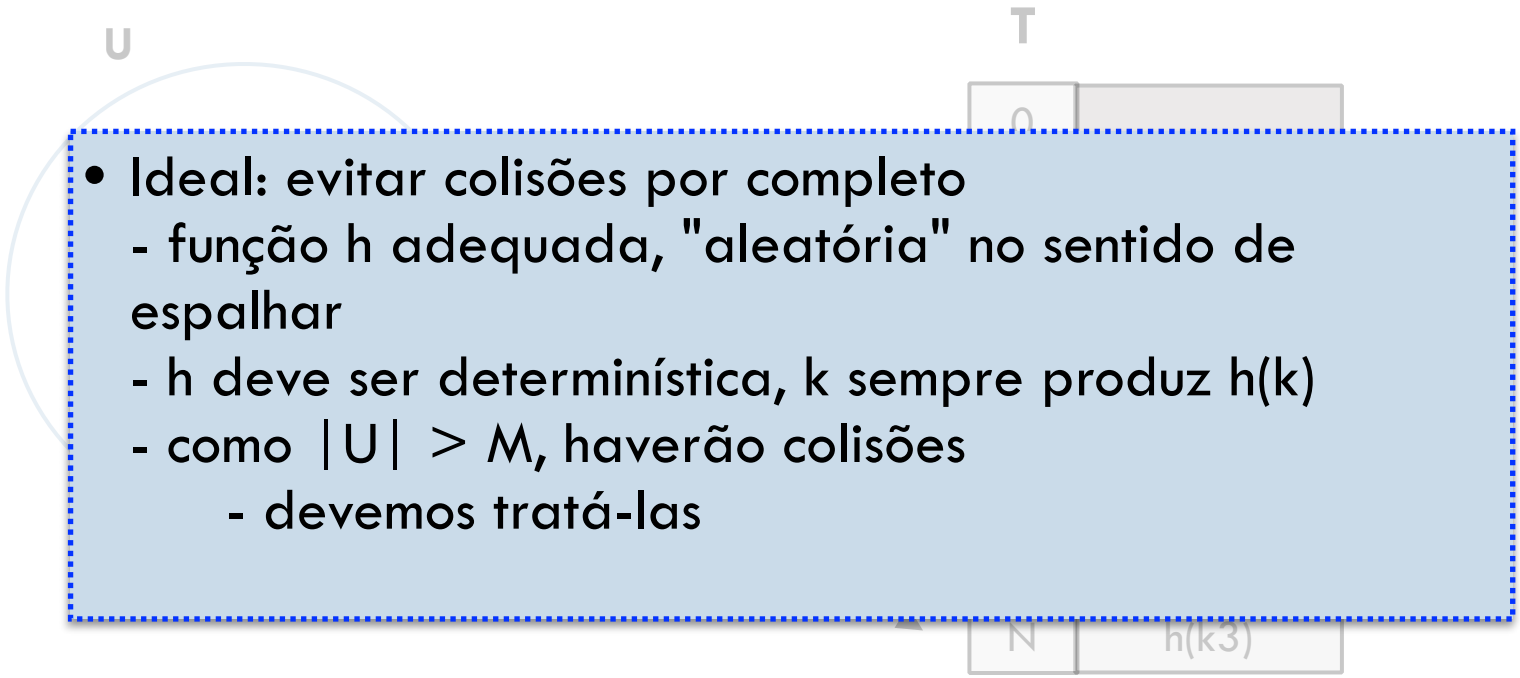


Tabela de Espalhamento



Colisão: duas ou mais chaves mapeadas para a mesma posição

Tabela de Espalhamento



- Ideal: evitar colisões por completo
  - função h adequada, "aleatória" no sentido de espalhar
  - h deve ser determinística, k sempre produz h(k)
  - como |U| > M, haverão colisões
    - devemos tratá-las

Colisão: duas ou mais chaves mapeadas para a mesma posição

Roteiro

1. Introdução
2. Tabelas de Endereçamento Direto
3. Tabelas de Espalhamento
4. **Resolução de colisões**
5. Funções Hash
6. Referências

Resolução de Colisões

A. Endereçamento Aberto

B. Resolução por encadeamento

Resolução de Colisões

A. Endereçamento Aberto

B. Resolução por encadeamento

Endereçamento aberto

"Quando uma chave colide com outra, a colisão é resolvida encontrando-se uma entrada diferente, e disponível"

- Se h(k) está ocupada, verifica:

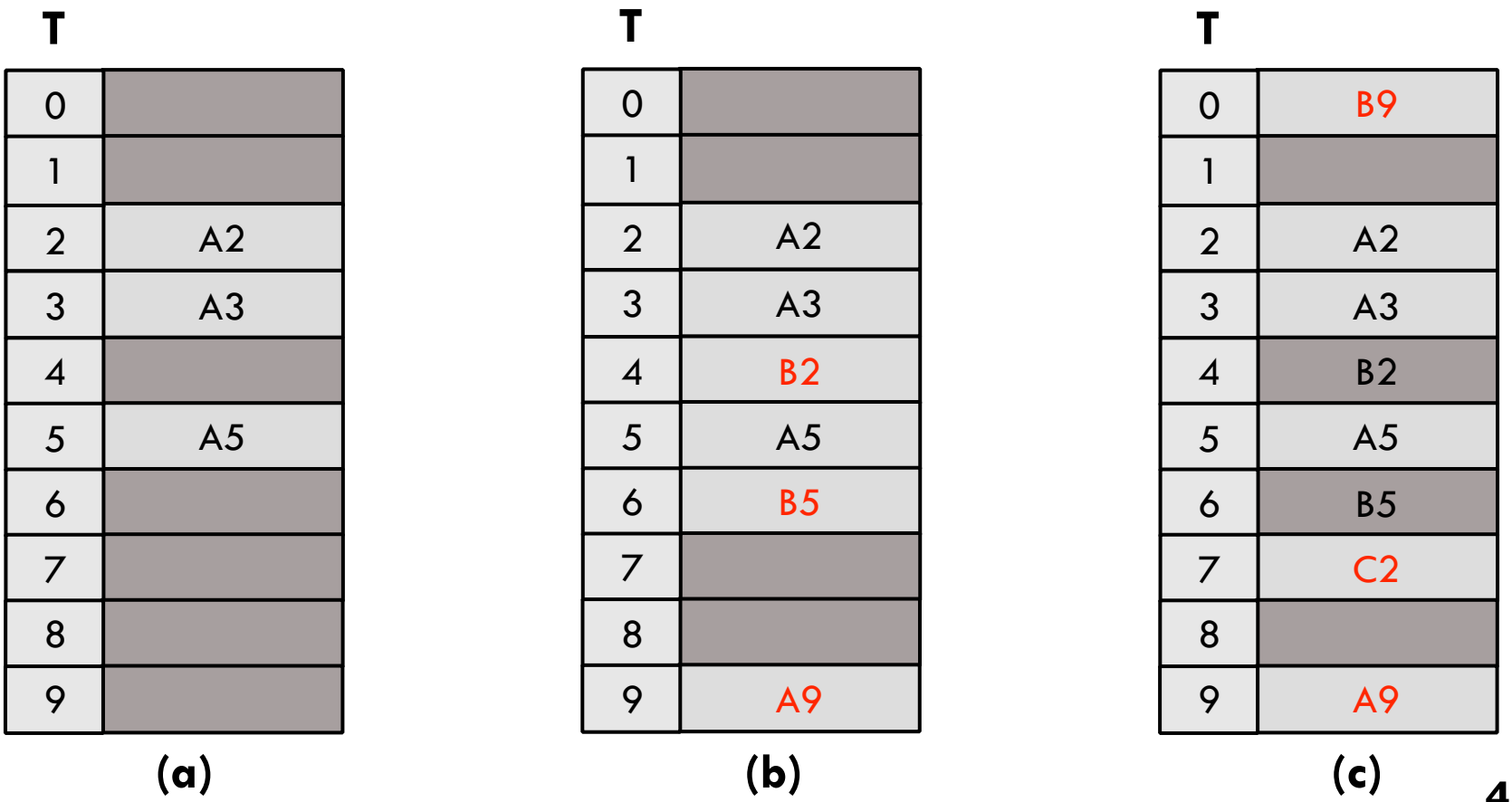
  … , h(k) + 1, h(k) + 2, h(k) + 3, …

- **Sondagem linear**

Exemplo: A2, A3, A5, B2, B5, A9, C2, B9
Tabela Hash de 10 espaços

Endereçamento aberto

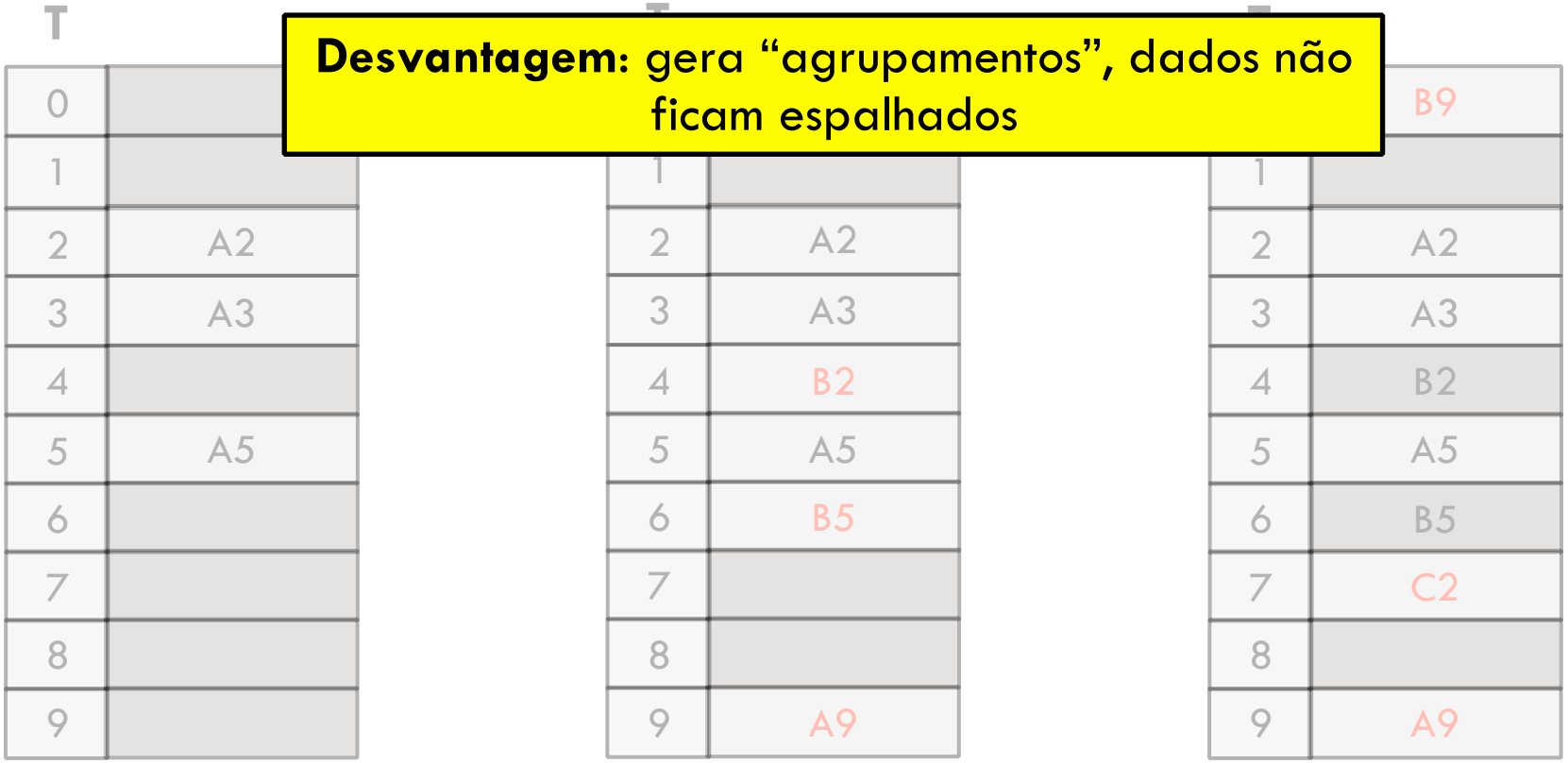
Ex: A2, A3, A5, B2, B5, A9, C2, B9



Endereçamento aberto

Ex: A2, A3, A5, B2, B5, A9, C2, B9

Desvantagem: gera "agrupamentos", dados não ficam espalhados



Endereçamento aberto

Alternativa: Sondagem quadrática

$h(k) + i^2, h(k) - i^2$ para $i = 1, 2, \ldots, (M-1)/2$

Sequencia de sondagens:

$h(k), h(k) + 1, h(k) - 1, h(k) + 4, h(k) - 4, h(k) + 9, h(k) - 9, \ldots$

Ex: A2, A3, A5, B2, B5, A9, C2, B9

Exercício 02

Ex: A2, A3, A5, B2, B5, A9, C2, B9 → Sondagem Quadrática

T

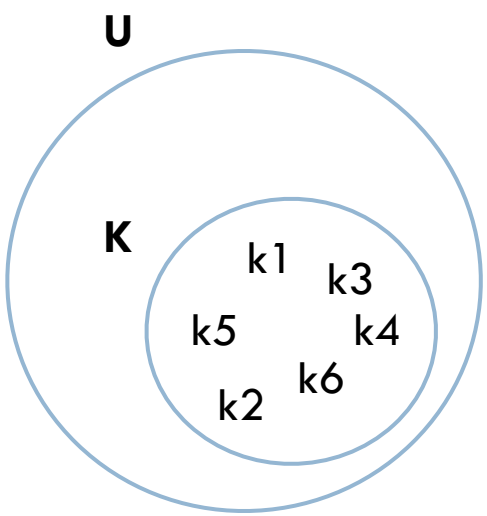
(a) **(b)** **(c)**

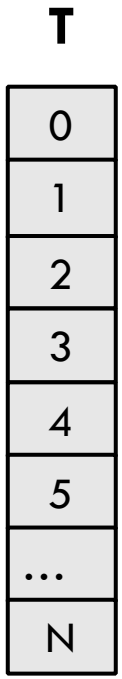
Resolução de Colisões

A. Endereçamento Aberto

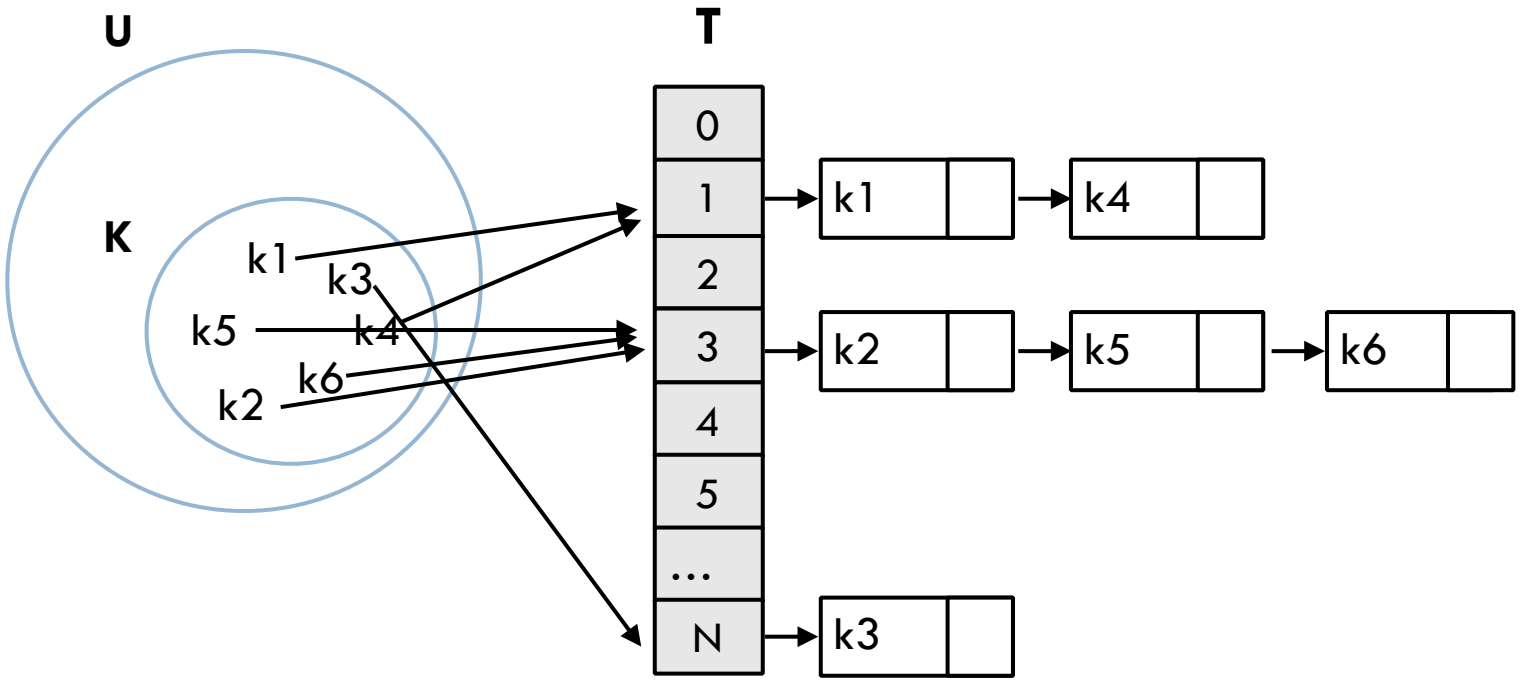
B. Resolução por encadeamento

Resolução por encadeamento

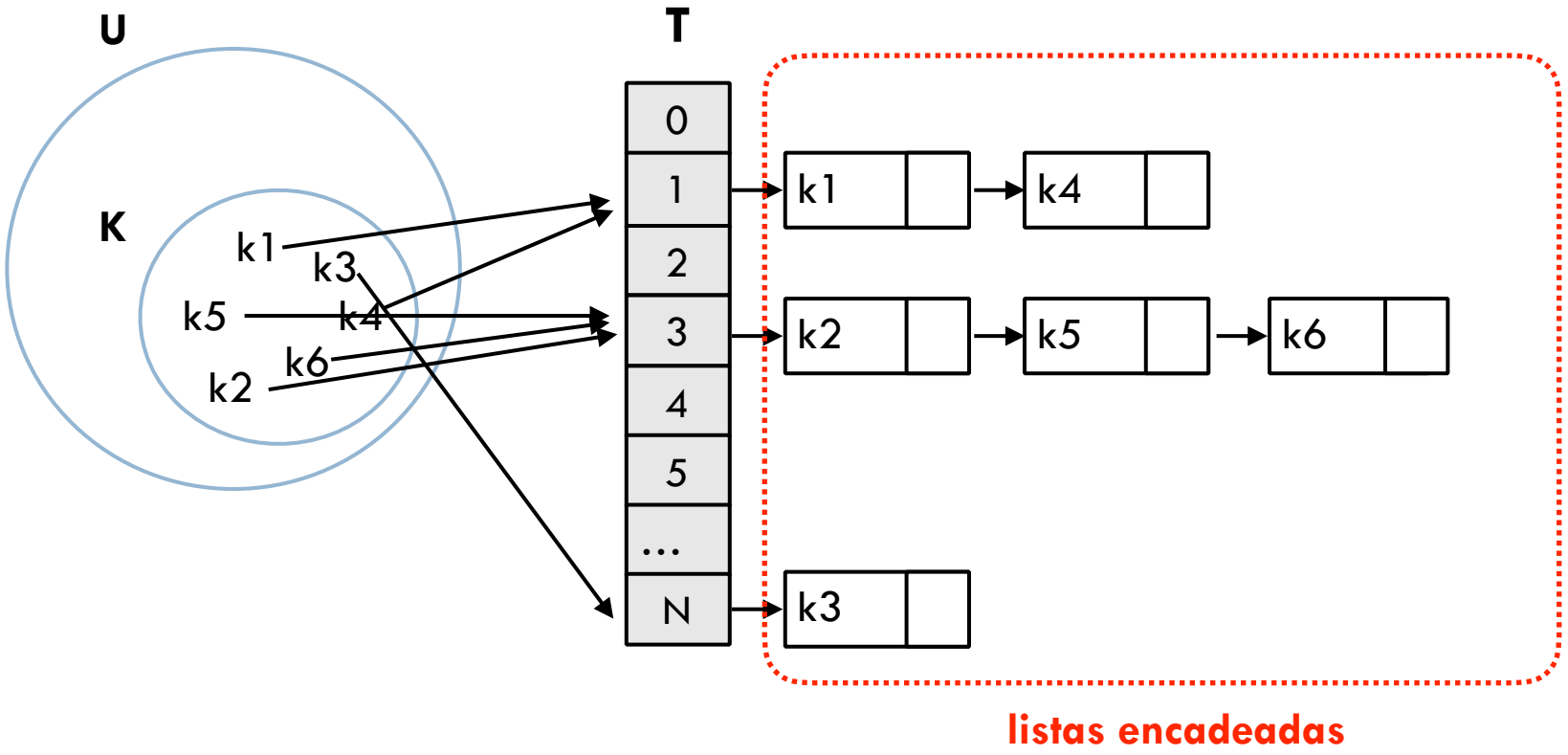




Resolução por encadeamento

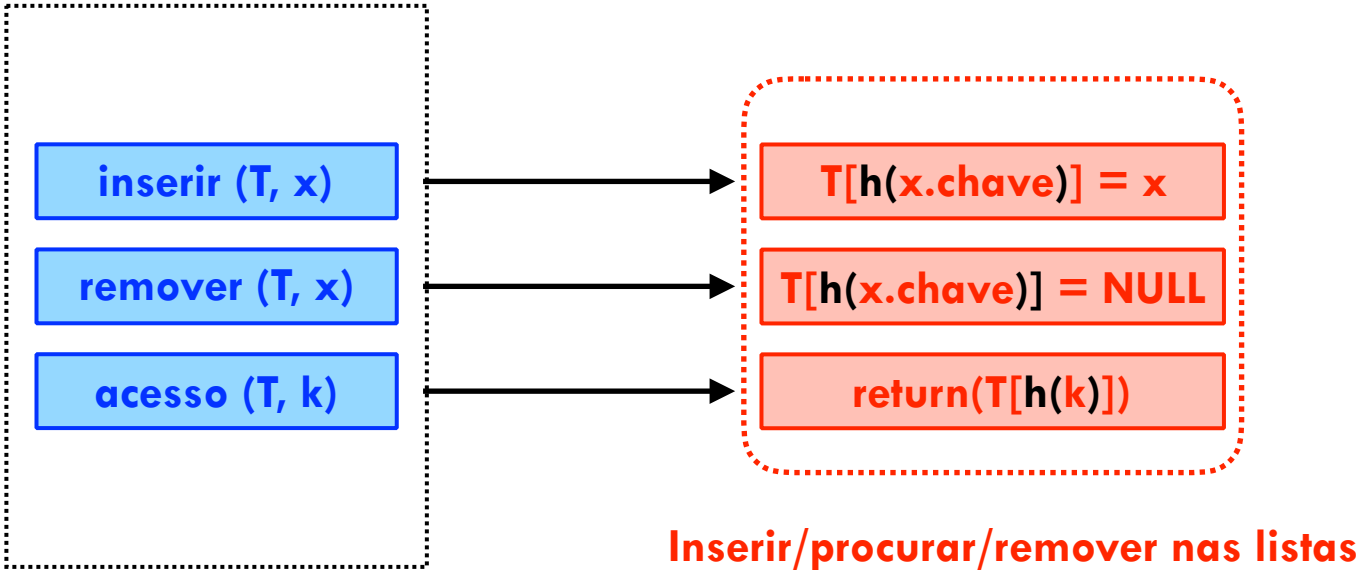


Resolução por encadeamento



Resolução por encadeamento

- **Operações de Modificação**



Inserir/procurar/remover nas listas indexadas pela posição da tabela

Roteiro

1. Introdução
2. Tabelas de Endereçamento Direto
3. Tabelas de Espalhamento
4. Resolução de colisões
5. **Funções Hash**
6. Referências

Funções Hash

Uma boa função deve:

- distribuir as chaves com igual probabilidade nas M posições
- na prática não se pode garantir, pois não temos ideia da distribuição de probabilidade das chaves
- usamos então **heurísticas**

A **Método da divisão**

B **Método da multiplicação**

Método da Divisão

Usa o resto da divisão de K por M:

$$h(k) = K \bmod M$$

"Valores de M"

* Evitar potência de 2
* Preferência por **números primos**

Exercício 03

Criar tabela Hash com M = 13, e inserir as chaves apresentadas abaixo. Use encadeamento para tratar das colisões.

k = 100	k = 40	k = 6
k = 0	k = 17	k = 15
k = 4	k = 25	k = 63
k = 48	k = 96	k = 2

Método da Multiplicação

Cria a função hash em duas etapas:

1) primeiro, multiplica K por uma constante A

$$0 < A < 1 \rightarrow (K * A)$$

2) multiplica por M, e toma o piso do resultado:

$$h(K) = \text{ground}(M * ((K * A) \bmod 1))$$

* Literatura sugere A = 0.618
* Pegar a parte inteira de h(k)

Exercício 04

Criar tabela Hash com M = 13 e A = 0.618, e inserir as chaves abaixo. Use encadeamento para tratar as colisões.

k = 100	k = 40	k = 6
k = 0	k = 17	k = 15
k = 4	k = 25	k = 63
k = 48	k = 96	k = 2

Exercício 05

Comparar as tabelas resultantes dos exercícios 3 e 4.

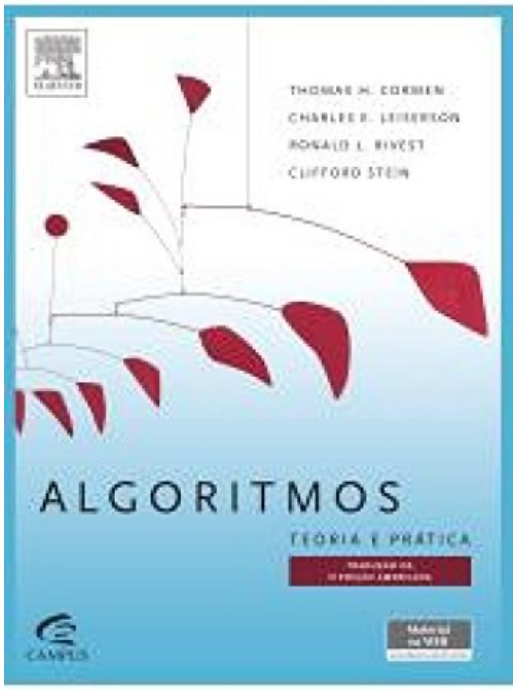
Exercício 06

Desenhe o conteúdo da tabela hash resultante da inserção de registros com as chaves: {A, G, U, D, E, S, L, I, C, H, P, R} nesta ordem, em uma tabela inicialmente vazia de tamanho 19 (dezenove), usando endereçamento aberto com hashing linear para a escolha de localizações alternativas. Use a função hash h(k) = k mod 19, para a k-ésima letra do alfabeto.

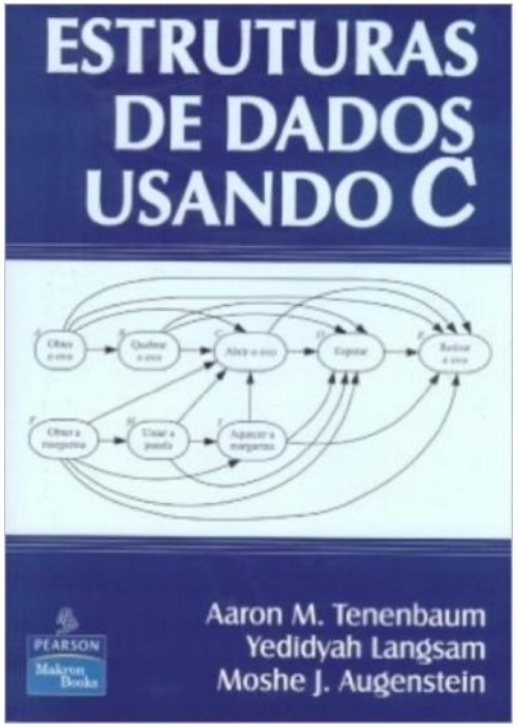
Roteiro

1. Introdução
2. Tabelas de Endereçamento Direto
3. Tabelas de Espalhamento
4. Resolução de colisões
5. Funções Hash
6. Referências

Referências sugeridas

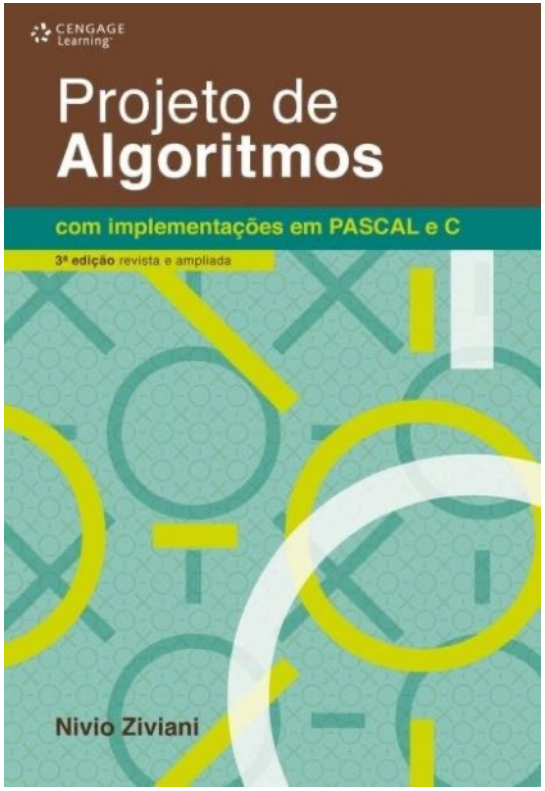


[Cormen et al, 2018]



[Tenenbaum et al, 1995]

Referências sugeridas



[Ziviani, 2010]

[Drozdek, 2017]

Perguntas?

Prof. Rafael G. **Mantovani**

rafaelmantovani@utfpr.edu.br