Aula 6 Tabela de Dispersão Tratamento de Colisão por Encadeamento

Estruturas de Dados Avançadas

Professor Eurinardo Rodrigues Costa Universidade Federal do Ceará Campus Russas

2021.1



Sumário

Prof.

Aula Passada

- Tratamento de Colisão por Encadeamento
 - Encadeamento Exterior
 - Encadeamento Interior
 - Fm duas Zonas
 - Em uma única zona

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeament Exterior

Encadeament

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadea-mento

Exterior Encadeamento Interior

Em duas Zona

Tabela de Dispersão (Hash)

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeamento Exterior Encadeamento Interior

Em duas Zonas

Tabela de Dispersão (Hash)

Acesso Direto

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Exterior
Encadeamento

Em duas Zonas

Tabela de Dispersão (Hash)

- Acesso Direto
- Funções de Dispersão (Hash)

FDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeament

Encadeamei

- - -

Em duas Zona

- Tabela de Dispersão (Hash)
 - Acesso Direto
 - Funções de Dispersão (Hash)
 - Método da Divisão

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadea-mento

Encadeamen

Encodoomo

Interior

Em duas Zonas

Tabela de Dispersão (Hash)

- Acesso Direto
- Funções de Dispersão (Hash)
 - Método da Divisão
 - Método da Dobra

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadea-mento

Encadeamen

Encodoomo

Interior

Em duas Zonas

- Tabela de Dispersão (Hash)
 - Acesso Direto
 - Funções de Dispersão (Hash)
 - Método da Divisão
 - Método da Dobra
 - Método da Multiplicação

EDA - Aula 6

Eurinardo

Aula Passada

Tratamento de Colisão por Encadea-mento

Encadeamen

Encadeame

....

Em duas Zonas

Tabela de Dispersão (Hash)

- Acesso Direto
- Funções de Dispersão (Hash)
 - Método da Divisão
 - Método da Dobra
 - Método da Multiplicação
 - Método da Análise de Dígitos

EDA - Aula 6

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeamen Exterior

Encadeame

Em duna Zanas

Em duas Zonas

- Tabela de Dispersão (Hash)
 - Acesso Direto
 - Funções de Dispersão (Hash)
 - Método da Divisão
 - Método da Dobra
 - Método da Multiplicação
 - Método da Análise de Dígitos
 - Problema de Colisão

FDA - Aula 6

Prof.

Aula Passada

Tratamento de Colisão por Encadea-

Encadeamento

Encadeament

Interior

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadea-

Encadeame

Exterior

Interior

Em uma única zona

Encadeamento Exterior

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeamer Exterior

Encadeam

- - -

Em uma única zon

Encadeamento Exterior

Simples aplicação de listas encadeadas em cada espaço da tabela.

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeamei Exterior

Encadoama

Interior

Em uma única a

Encadeamento Exterior

Simples aplicação de listas encadeadas em cada espaço da tabela.

Complexidade no pior caso = O(n).

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeamen Exterior

Encadeame

Em duas Zonas

Em duas Zonas

Encadeamento Exterior

Simples aplicação de listas encadeadas em cada espaço da tabela.

Complexidade no pior caso = O(n).

Encadeamento Interior

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeamento Exterior Encadeamento

Interior Em duas Zonas

Encadeamento Exterior

Simples aplicação de listas encadeadas em cada espaço da tabela.

Complexidade no pior caso = O(n).

Encadeamento Interior

As listas encadeadas compartilham a mesma memória da tabela. Temos:

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeamento Exterior Encadeamento

Interior Em duas Zonas

Encadeamento Exterior

Simples aplicação de listas encadeadas em cada espaço da tabela.

Complexidade no pior caso = O(n).

Encadeamento Interior

As listas encadeadas compartilham a mesma memória da tabela. Temos:

em duas zonas

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeamento Exterior Encadeamento

Interior
Em duas Zonas

Encadeamento Exterior

Simples aplicação de listas encadeadas em cada espaço da tabela.

Complexidade no pior caso = O(n).

Encadeamento Interior

As listas encadeadas compartilham a mesma memória da tabela. Temos:

- em duas zonas e
- em uma única zona.

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadea-

Encadeament Exterior

Encadeame

Em duas Zonas

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento d Colisão por Encadeamento

Encadeamer Exterior

Encadeame

Em duas Zonas

Em uma única zona

Em duas zonas

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Exterior ____

Interior

Em duas Zonas

Em duas zonas

A tabela é dividida em zona de endereços e zona de colisões.

EDA - Aula 6

Prof. Eurinardo

Aula Passad

Tratamento de Colisão por Encadea-mento

Encadeament Exterior

Interior

Em duas Zonas

Em duas zonas

A tabela é dividida em zona de endereços e zona de colisões.

Problema: pode ocorrer falso overflow!!!

EDA - Aula 6

Prof. Eurinardo

Aula Passad

Tratamento de Colisão por Encadeamento

Encadeamento Exterior

Interior

Em duas Zonas

Em duas zonas

A tabela é dividida em zona de endereços e zona de colisões.

Problema: pode ocorrer falso overflow!!!

EDA - Aula 6

Prof. Eurinardo

Aula Passad

Tratamento de Colisão por Encadeamento

Encadeamento Exterior Encadeamento

Em duas Zonas

Em uma única zo

Em duas zonas

A tabela é dividida em zona de endereços e zona de colisões.

Problema: pode ocorrer falso overflow!!!

Em uma única zona

Uma única zona para endereços e colisões.

EDA - Aula 6

Prof. Eurinardo

Aula Passad

Tratamento de Colisão por Encadeamento

Encadeamento Exterior Encadeamento Interior

Em duas Zonas

Em duas Zonas Em uma única zon

Em duas zonas

A tabela é dividida em zona de endereços e zona de colisões.

Problema: pode ocorrer falso overflow!!!

- Uma única zona para endereços e colisões.
- Cada elemento da tabela está em uma lista circula.

EDA - Aula 6

Prof. Eurinardo

Aula Passad

Tratamento de Colisão por Encadeamento

Exterior
Encadeamento
Interior

Em duas Zonas

Em duas zonas

A tabela é dividida em zona de endereços e zona de colisões.

Problema: pode ocorrer falso overflow!!!

- Uma única zona para endereços e colisões.
- Cada elemento da tabela está em uma lista circula.
- O início da tabela, temos apenas listas circulares.

Em duas Zonas

Em duas zonas

A tabela é dividida em zona de endereços e zona de colisões.

Problema: pode ocorrer falso overflow!!!

- Uma única zona para endereços e colisões.
- Cada elemento da tabela está em uma lista circula.
- O início da tabela, temos apenas listas circulares.
- Considere os campos:

Em duas Zonas

Em duas zonas

A tabela é dividida em zona de endereços e zona de colisões.

Problema: pode ocorrer falso overflow!!!

- Uma única zona para endereços e colisões.
- Cada elemento da tabela está em uma lista circula.
- O início da tabela, temos apenas listas circulares.
- Considere os campos: chave.

FDA - Aula 6

Prof. Eurinardo

Aula Passad

Tratamento de Colisão por Encadeamento

Encadeamento Exterior Encadeamento Interior

Em duas Zonas

Em duas zonas

A tabela é dividida em zona de endereços e zona de colisões.

Problema: pode ocorrer falso overflow!!!

- Uma única zona para endereços e colisões.
- Cada elemento da tabela está em uma lista circula.
- O início da tabela, temos apenas listas circulares.
- Considere os campos: chave, estado

EDA - Aula 6

Prof. Eurinardo

Aula Passad

Tratamento de Colisão por Encadeamento

Exterior
Encadeamento
Interior

Em duas Zonas

Em duas zonas

A tabela é dividida em zona de endereços e zona de colisões.

Problema: pode ocorrer falso overflow!!!

- Uma única zona para endereços e colisões.
- Cada elemento da tabela está em uma lista circula.
- O início da tabela, temos apenas listas circulares.
- Considere os campos: chave, estado (ocupado, não ocupado)

FDA - Aula 6

Prof. Eurinardo

Aula Passad

Tratamento de Colisão por Encadeamento

Encadeamento Exterior Encadeamento Interior

Em duas Zonas

Em duas zonas

A tabela é dividida em zona de endereços e zona de colisões.

Problema: pode ocorrer falso overflow!!!

- Uma única zona para endereços e colisões.
- Cada elemento da tabela está em uma lista circula.
- O início da tabela, temos apenas listas circulares.
- Considere os campos: chave, estado (ocupado, não ocupado) e pont

FDA - Aula 6

Prof. Eurinardo

Aula Passad

Tratamento de Colisão por Encadeamento

Exterior
Encadeamento
Interior

Em duas Zonas

Em duas zonas

A tabela é dividida em zona de endereços e zona de colisões.

Problema: pode ocorrer falso overflow!!!

- Uma única zona para endereços e colisões.
- Cada elemento da tabela está em uma lista circula.
- O início da tabela, temos apenas listas circulares.
- Considere os campos: chave, estado (ocupado, não ocupado) e pont (ponteiro para o próximo)

Busca em uma única zona

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeament

Exterior

- - -

Em uma única zona

Algoritmo 1: Buscar(x, end, a)

Busca em uma única zona

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeament Exterior

Encadeament

Em duna Zana

Em uma única zona

Algoritmo 2: Buscar(x, end, a)

Entrada: x: chave para ser encontrada

Busca em uma única zona

Prof. Eurinardo

Algoritmo 3: Buscar(x, end, a)

Entrada: x: chave para ser encontrada

Saída:

compart. não ocupado na lista h(x), caso exista

, C.C.

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadea-mento

Encadeamen

Encadeamen

Em duna Zana

Em uma única zona

Algoritmo 4: Buscar(x, end, a)

Entrada: x: chave para ser encontrada

Saída:

$$j = \begin{cases} \text{compart. não ocupado na lista } h(x) &, \text{ caso exista} \\ \lambda &, \text{ c.c.} \end{cases}$$

$$a = \begin{cases} 1 &, \text{ se } x \text{ for encontrado} \\ 2 &, \text{ caso contrário} \end{cases}$$

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeamen

Encadeamer

Em duna Zana

Em uma única zona

```
Algoritmo 5: Buscar(x, end, a)
```

Entrada: x: chave para ser encontrada Saída:

$$j = \begin{cases} \text{compart. não ocupado na lista } h(x) &, \text{ caso exista} \\ \lambda &, \text{ c.c.} \end{cases}$$

$$a = \begin{cases} 1 &, \text{ se } x \text{ for encontrado} \\ 2 &, \text{ caso contrário} \end{cases}$$

$$\text{end} = \begin{cases} j &, \text{ se chave não encontrada} \\ \text{endereço de } x &, \text{ caso contrário} \end{cases}$$

1 *a* ← 0:

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadea-mento

Encadeamen

Encadeamer

Em duas Zona

```
Algoritmo 6: Buscar(x, end, a)

Entrada: x: chave para ser encontrada

Saída:
j = \begin{cases} \text{compart. não ocupado na lista } h(x) &, \text{ caso exista} \\ \lambda &, \text{ c.c.} \end{cases}
a = \begin{cases} 1 &, \text{ se } x \text{ for encontrado} \\ 2 &, \text{ caso contrário} \end{cases}
\text{end} = \begin{cases} j &, \text{ se chave não encontrada} \\ \text{endereço de } x &, \text{ caso contrário} \end{cases}
```

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadea-mento

Encadeamen

Encadeamen

Em duna Zana

```
Algoritmo 7: Buscar(x, end, a)
Entrada: x: chave para ser encontrada
Saída:
j = \begin{cases} \text{compart. não ocupado na lista } h(x) &, \text{ caso exista} \\ \lambda &, \text{ c.c.} \end{cases}
a = \begin{cases} 1 &, \text{ se } x \text{ for encontrado} \\ 2 &, \text{ caso contrário} \end{cases}
\text{end} = \begin{cases} j &, \text{ se chave não encontrada} \\ \text{ endereço de } x &, \text{ caso contrário} \end{cases}
1 \quad a \leftarrow 0; \text{ end } \leftarrow h(x);
```

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadea-mento

Encadeamen

Encadeamer

Em duas Zona

```
Algoritmo 8: Buscar(x, end, a)

Entrada: x: chave para ser encontrada

Saída:
j = \begin{cases} \text{compart. não ocupado na lista } h(x) &, \text{ caso exista} \\ \lambda &, \text{ c.c.} \end{cases}
a = \begin{cases} 1 &, \text{ se } x \text{ for encontrado} \\ 2 &, \text{ caso contrário} \end{cases}
\text{end} = \begin{cases} j &, \text{ se chave não encontrada} \\ \text{endereço de } x &, \text{ caso contrário} \end{cases}
```

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadea-mento

Encadeamen

Encodoomo

Em duas Zona

Em uma única zona

```
Algoritmo 9: Buscar(x, end, a)
```

Entrada: x: chave para ser encontrada Saída:

```
j = \begin{cases} \text{compart. não ocupado na lista } h(x) &, \text{ caso exista} \\ \lambda &, \text{ c.c.} \end{cases}
a = \begin{cases} 1 &, \text{ se } x \text{ for encontrado} \\ 2 &, \text{ caso contrário} \end{cases}
\text{end} = \begin{cases} j &, \text{ se chave não encontrada} \\ \text{endereço de } x &, \text{ caso contrário} \end{cases}
```

- 1 $a \leftarrow 0$; end $\leftarrow h(x)$; $j \leftarrow \lambda$;
- 2 enquanto a=0 faça

```
EDA - Aula 6
```

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeamen

Encodoomo

IIIterioi

EIII uuds Zuilas

```
Algoritmo 10: Buscar(x, end, a)
    Entrada: x: chave para ser encontrada
   Saída:
            compart. não ocupado na lista h(x), caso exista
                                                                        , C.C.
   a = \begin{cases} 1 & \text{, se } x \text{ for encontrado} \\ 2 & \text{, caso contrário} \end{cases}
   \mathsf{end} = \begin{cases} j & \text{, se chave não encontrada} \\ \mathsf{endereço} \ \mathsf{de} \ x & \text{, caso contrário} \end{cases}
1 a \leftarrow 0; end \leftarrow h(x); j \leftarrow \lambda;
2 enquanto a = 0 faça
           se T[end].estado = não ocupado então
3
```

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeamen

Encadoamo

. . .

```
Algoritmo 11: Buscar(x, end, a)
   Entrada: x: chave para ser encontrada
   Saída:
           compart. não ocupado na lista h(x), caso exista
                                                                     , C.C.
  a = \begin{cases} 1 & \text{, se } x \text{ for encontrado} \\ 2 & \text{, caso contrário} \end{cases}
  end = \begin{cases} j & \text{, se chave não encontrada} \\ endereço de x & \text{, caso contrário} \end{cases}
1 a \leftarrow 0; end \leftarrow h(x); j \leftarrow \lambda;
   enquanto a = 0 faça
          se T[end].estado = não ocupado então
3
                i \leftarrow \text{end}
4
```

```
EDA - Aula 6
```

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeamen

Encadeame

Em duae Zone

```
Algoritmo 12: Buscar(x, end, a)
   Entrada: x: chave para ser encontrada
   Saída:
           compart. não ocupado na lista h(x), caso exista
                                                                     , C.C.
  a = \begin{cases} 1 & \text{, se } x \text{ for encontrado} \\ 2 & \text{, caso contrário} \end{cases}
   \mathsf{end} = \begin{cases} j & \text{, se chave não encontrada} \\ \mathsf{endereço} \ \mathsf{de} \ x & \text{, caso contrário} \end{cases}
1 a \leftarrow 0; end \leftarrow h(x); j \leftarrow \lambda;
   enquanto a = 0 faça
          se T[end].estado = não ocupado então
3
                i \leftarrow \text{end}
4
          se T[end].chave = x e T[end].estado = ocupado
5
            então
```

```
EDA - Aula 6
```

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeamen

Encadeame

Em duas Zona

```
Algoritmo 13: Buscar(x, end, a)
   Entrada: x: chave para ser encontrada
   Saída:
          compart. não ocupado na lista h(x), caso exista
                                                                 , C.C.
  a = \begin{cases} 1 & \text{, se } x \text{ for encontrado} \\ 2 & \text{, caso contrário} \end{cases}
  end = \begin{cases} j & \text{, se chave não encontrada} \\ endereço de x & \text{, caso contrário} \end{cases}
1 a \leftarrow 0; end \leftarrow h(x); j \leftarrow \lambda;
  enquanto a = 0 faça
         se T[end].estado = não ocupado então
3
               i \leftarrow \text{end}
4
         se T[end].chave = x e T[end].estado = ocupado
5
           então
                a ← 1
6
```

```
EDA - Aula 6
```

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadea-mento

Encadeamen

Encadeame

Em duna Zana

```
Algoritmo 14: Buscar(x, end, a)
   Entrada: x: chave para ser encontrada
   Saída:
           compart. não ocupado na lista h(x), caso exista
                                                                   , C.C.
  a = \begin{cases} 1 & \text{, se } x \text{ for encontrado} \\ 2 & \text{, caso contrário} \end{cases}
  \mathsf{end} = \begin{cases} j & \text{, se chave não encontrada} \\ \mathsf{endereço} \ \mathsf{de} \ x & \text{, caso contrário} \end{cases}
1 a \leftarrow 0; end \leftarrow h(x); j \leftarrow \lambda;
  enquanto a = 0 faça
         se T[end].estado = não ocupado então
3
                i \leftarrow \text{end}
4
         se T[end].chave = x e T[end].estado = ocupado
5
           então
                a ← 1 %chave encontrada
6
```

```
EDA - Aula 6
```

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeamen

Encadoamo

Em duae Zona

- ...

```
Algoritmo 15: Buscar(x, end, a)
   Entrada: x: chave para ser encontrada
   Saída:
           compart. não ocupado na lista h(x), caso exista
                                                                  , C.C.
  a = \begin{cases} 1 & \text{, se } x \text{ for encontrado} \\ 2 & \text{, caso contrário} \end{cases}
  \mathsf{end} = \begin{cases} j & \text{, se chave não encontrada} \\ \mathsf{endereço} \ \mathsf{de} \ x & \text{, caso contrário} \end{cases}
1 a \leftarrow 0; end \leftarrow h(x); j \leftarrow \lambda;
  enquanto a = 0 faça
         se T[end].estado = não ocupado então
3
                i \leftarrow \text{end}
4
         se T[end].chave = x e T[end].estado = ocupado
5
           então
                a ← 1 %chave encontrada
6
7
         senão
```

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeamen

Encadeame

Em duas Zona

```
Algoritmo 16: Buscar(x, end, a)
   Entrada: x: chave para ser encontrada
   Saída:
          compart. não ocupado na lista h(x), caso exista
                                                                 , C.C.
  a = \begin{cases} 1 & \text{, se } x \text{ for encontrado} \\ 2 & \text{, caso contrário} \end{cases}
  \mathsf{end} = \begin{cases} j & \text{, se chave não encontrada} \\ \mathsf{endereço} \ \mathsf{de} \ x & \text{, caso contrário} \end{cases}
1 a \leftarrow 0; end \leftarrow h(x); j \leftarrow \lambda;
  enquanto a = 0 faça
         se T[end].estado = não ocupado então
3
               i \leftarrow \text{end}
4
         se T[end].chave = x e T[end].estado = ocupado
5
           então
                a ← 1 %chave encontrada
6
7
         senão
                end ← T[end].pont
8
```

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeamen

Encadeame

Em duna Zana

```
Algoritmo 17: Buscar(x, end, a)
   Entrada: x: chave para ser encontrada
   Saída:
          compart. não ocupado na lista h(x), caso exista
                                                                , C.C.
  a = \begin{cases} 1 & \text{, se } x \text{ for encontrado} \\ 2 & \text{, caso contrário} \end{cases}
  \mathsf{end} = \begin{cases} j & \text{, se chave não encontrada} \\ \mathsf{endereço} \ \mathsf{de} \ x & \text{, caso contrário} \end{cases}
1 a \leftarrow 0; end \leftarrow h(x); j \leftarrow \lambda;
  enquanto a = 0 faça
         se T[end].estado = não ocupado então
3
               i \leftarrow \text{end}
4
         se T[end].chave = x e T[end].estado = ocupado
5
           então
                a ← 1 %chave encontrada
6
7
         senão
               end ← T[end].pont
8
                se end = h(x) então
9
```

```
EDA - Aula 6
```

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeamen

Encadeame

Em duas Zona

```
Algoritmo 18: Buscar(x, end, a)
    Entrada: x: chave para ser encontrada
    Saída:
            compart. não ocupado na lista h(x), caso exista
                                                                   , C.C.
   a = \begin{cases} 1 & \text{, se } x \text{ for encontrado} \\ 2 & \text{, caso contrário} \end{cases}
   \mathsf{end} = \begin{cases} j & \text{, se chave não encontrada} \\ \mathsf{endereço} \ \mathsf{de} \ x & \text{, caso contrário} \end{cases}
1 a \leftarrow 0; end \leftarrow h(x); j \leftarrow \lambda;
   enquanto a = 0 faça
          se T[end].estado = não ocupado então
3
                 i \leftarrow \text{end}
 4
          se T[end].chave = x e T[end].estado = ocupado
5
            então
                 a ← 1 %chave encontrada
6
          senão
7
                 end \leftarrow T[end].pont
 8
                 se end = h(x) então
 9
                       a \leftarrow 2; end \leftarrow i;
10
```

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadea-mento

Encadeamen

Encadeame

Em duna Zana

```
Algoritmo 19: Buscar(x, end, a)
    Entrada: x: chave para ser encontrada
    Saída:
           compart. não ocupado na lista h(x), caso exista
                                                                  , C.C.
   a = \begin{cases} 1 & \text{, se } x \text{ for encontrado} \\ 2 & \text{, caso contrário} \end{cases}
   \mathsf{end} = \begin{cases} j & \text{, se chave não encontrada} \\ \mathsf{endereço} \ \mathsf{de} \ x & \text{, caso contrário} \end{cases}
1 a \leftarrow 0; end \leftarrow h(x); j \leftarrow \lambda;
   enquanto a = 0 faça
          se T[end].estado = não ocupado então
3
                i \leftarrow \text{end}
 4
          se T[end].chave = x e T[end].estado = ocupado
5
            então
                 a ← 1 %chave encontrada
6
          senão
7
                 end \leftarrow T[end].pont
 8
                 se end = h(x) então
 9
                       a \leftarrow 2; end \leftarrow j; %não chave encontrada
10
```

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento d Colisão por Encadeamento

Encadeamen

Encadeamen

Em duas Zona

Em uma única zona

Algoritmo 20: Remover(x, end, a)

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeamen Exterior

Encadeamen

Em duae Zona

Em uma única zor

Algoritmo 21: Remover(x, end, a)

1 Buscar(x, end, a)

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeament

Exterior

Interior

Em duas Zonas

Algoritmo 22: Remover(x, end, a)

1 Buscar(x, end, a)

2 se a = 1 então

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeamen

Exterior

Interior

Em duas Zonas

Algoritmo 23: Remover(x, end, a)

- 1 Buscar(x, end, a)
- 2 se a = 1 então
 - T[end].estado ← não ocupado

EDA - Aula 6

Prof. Eurinardo

Aula Passad

Tratamento de Colisão por Encadeamento

Encadeament

Exterior

Interior

Em duas Zonas

Em uma única zona

Algoritmo 24: Remover(x, end, a)

- 1 Buscar(x, end, a)
- 2 se a=1 então
- 3 T[end].estado ← não ocupado
- 4 senão

EDA - Aula 6

Prof. Eurinardo

Aula Passad

Tratamento de Colisão por Encadeamento

Encadeamen

EXIGNO

Interior

Em duas Zonas

Em uma única zona

Algoritmo 25: Remover(x, end, a)

- 1 Buscar(x, end, a)
- 2 se a=1 então
- 3 T[end].estado ← não ocupado
- 4 senão
- 5 "Remoção inválida"

Algoritmo 26: Remover(x, end, a)

- Buscar(x, end, a)
- 2 se a=1 então
- T[end].estado ← não ocupado
- senão
- "Remoção inválida"

EDA - Aula 6

Prof. Eurinardo

Aula Passad

Tratamento de Colisão por Encadeamento

Encadeamen Exterior

Interior

Em duas Zonas Em uma única zona Algoritmo 27: Inserir(x, end, a)

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeament

Exterior

IIItorioi

Em uma única zona

Algoritmo 28: Inserir(x, end, a)

1 Buscar(x, end, a)

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeamen

Encadoamo

Em duna Zana

Em uma única zona

Algoritmo 29: Inserir(x, end, a)

- 1 Buscar(x, end, a)
- 2 se a = 1 então

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeamen Exterior

Encadeame

Em duna Zana

Em uma única zona

Algoritmo 30: Inserir(x, end, a)

- 1 Buscar(x, end, a)
- 2 se a = 1 então "Inserção inválida: chave já existe"

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadea-

Encadeamen

Encadoam

Em uma única zona

Algoritmo 31: Inserir(x, end, a)

- 1 Buscar(x, end, a)
- 2 **se** a = 1 **então** "Inserção inválida: chave já existe";
- 3 senão

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeamen Exterior

Encadeame

Em duna Zana

Liii dada Lorida

Algoritmo 32: Inserir(x, end, a)

- 1 Buscar(x, end, a)
- 2 **se** a=1 **então** "Inserção inválida: chave já existe";
- 3 senão
 - se $end \neq \lambda$ então

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeamen Exterior

Encadeame

Em duas Zonas

Algoritmo 33: Inserir(x, end, a)

- 1 Buscar(x, end, a)
- 2 **se** a=1 **então** "Inserção inválida: chave já existe";
- 3 senão
- **se** $end \neq \lambda$ **então** $j \leftarrow$ end

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeamen

Encadoame

III(GIIOI

Em duas Zona

```
Algoritmo 34: Inserir(x, end, a)
```

```
1 Buscar(x, end, a)
```

- 2 **se** a=1 **então** "Inserção inválida: chave já existe";
- 3 senão
- **se** $end \neq \lambda$ **então** $j \leftarrow$ end ;
- 5 senão

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadea-

Encadeamen

Encadoamo

_ _ _

LIII dudo Zurido

```
Algoritmo 35: Inserir(x, end, a)
```

```
1 Buscar(x, end, a)
2 se a=1 então "Inserção inválida: chave já existe";
3 senão
4 se end \neq \lambda então j \leftarrow end;
5 senão
6 i \leftarrow 1;
```

5

6

Prof.

```
Algoritmo 36: Inserir(x, end, a)
```

```
1 Buscar(x, end, a)
2 se a=1 então "Inserção inválida: chave já existe";
3 senão
        se end \neq \lambda então j \leftarrow end ;
        senão
             i \leftarrow 1; j \leftarrow h(x)
```

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadea-

Encadeamen

Encodoomo

Interior

Em duas Zona

Algoritmo 37: Inserir(x, end, a)

```
1 Buscar(x, end, a)
2 se a=1 então "Inserção inválida: chave já existe";
3 senão
4 se end \neq \lambda então j \leftarrow end ;
5 senão
6 i \leftarrow 1; j \leftarrow h(x)
7 enquanto i \leq m faça
```

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadea-

Encadeamen

Interior

Em duas Zona

```
Algoritmo 38: Inserir(x, end, a)
```

```
1 Buscar(x, end, a)
2 se a=1 então "Inserção inválida: chave já existe";
3 senão
        se end \neq \lambda então j \leftarrow end ;
        senão
5
             i \leftarrow 1; j \leftarrow h(x)
6
             enquanto i \leq m faça
                  se t[j].estado = ocupado então
8
```

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadea-

Encadeamen

Encadeament

Em duas Zona

```
Algoritmo 39: Inserir(x, end, a)
```

```
1 Buscar(x, end, a)
2 se a=1 então "Inserção inválida: chave já existe";
3 senão
4 se end \neq \lambda então j \leftarrow end;
5 senão
6 i \leftarrow 1; j \leftarrow h(x)
enquanto i \leq m faça
8 se t[j].estado = ocupado então
9 j \leftarrow (j+1) \mod m
```

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeamen

Encadeamento

Em duas Zonas

```
Algoritmo 40: Inserir(x, end, a)
```

```
1 Buscar(x, end, a)
2 se a=1 então "Inserção inválida: chave já existe";
3 senão
4 se end \neq \lambda então j \leftarrow end ;
5 senão
6 i \leftarrow 1; j \leftarrow h(x)
7 enquanto i \leq m faça
8 se t[j].estado = ocupado então
9 j \leftarrow (j+1) \mod m
10 i \leftarrow i+1
```

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadea-

Encadeamer

Interior

Em duas Zonas

```
Algoritmo 41: Inserir(x, end, a)
```

```
1 Buscar(x, end, a)
2 se a=1 então "Inserção inválida: chave já existe";
3 senão
        se end \neq \lambda então j \leftarrow end ;
        senão
5
              i \leftarrow 1; j \leftarrow h(x)
 6
              enquanto i \leq m faça
                   se t[j].estado = ocupado então
 8
                        j \leftarrow (j+1) \mod m
                        i \leftarrow i + 1
10
                   senão
11
```

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeamen

Encadeamer

Em duna Zana

EIII uuds Zuilds

Algoritmo 42: Inserir(x, end, a)

```
1 Buscar(x, end, a)
2 se a=1 então "Inserção inválida: chave já existe";
3 senão
         se end \neq \lambda então j \leftarrow end ;
         senão
5
              i \leftarrow 1; j \leftarrow h(x)
 6
              enquanto i \leq m faça
 7
                    se t[j].estado = ocupado então
 8
                         j \leftarrow (j+1) \mod m
                         i \leftarrow i + 1
10
                    senão
11
                         i \leftarrow m + 2
12
```

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeamen

Encadeament

Em duas Zonas

```
Algoritmo 43: Inserir(x, end, a)
```

```
1 Buscar(x, end, a)
2 se a=1 então "Inserção inválida: chave já existe";
3 senão
         se end \neq \lambda então j \leftarrow end ;
         senão
5
              i \leftarrow 1; j \leftarrow h(x)
 6
              enquanto i \leq m faça
 7
                    se t[j].estado = ocupado então
 8
                         j \leftarrow (j+1) \mod m
 9
                         i \leftarrow i + 1
10
                   senão
11
                        i \leftarrow m + 2
12
```

13

Algoritmo 44: Inserir(x, end, a)

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeamen

Enondoom

Em duas Zonas

Buscar(x, end, a) se a=1 então "Inserção inválida: chave já existe"; senão se $end \neq \lambda$ então $j \leftarrow$ end ;

```
senão
5
               i \leftarrow 1; j \leftarrow h(x)
 6
               enquanto i \leq m faça
 7
                     se t[i].estado = ocupado então
 8
                          j \leftarrow (j+1) \mod m
 9
                          i \leftarrow i + 1
10
                     senão
11
                        i \leftarrow m + 2
12
```

se i = m + 1 então "overflow";

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadea-

Encadeamen

III(GIIOI

Em duas Zonas

```
Algoritmo 45: Inserir(x, end, a)
```

```
1 Buscar(x, end, a)
2 se a=1 então "Inserção inválida: chave já existe";
3 senão
        se end \neq \lambda então j \leftarrow end ;
        senão
5
              i \leftarrow 1; j \leftarrow h(x)
 6
              enquanto i \leq m faça
 7
                   se t[i].estado = ocupado então
 8
                        j \leftarrow (j+1) \mod m
 9
                        i \leftarrow i + 1
10
                   senão
11
                      i \leftarrow m + 2
12
              se i = m + 1 então "overflow"; pare
13
```

14

15

16

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeamer

Encodoome

LIII duas Zurias

```
Algoritmo 46: Inserir(x, end, a)
Buscar(x, end, a)
2 se a = 1 então "Inserção inválida: chave já existe";
3 senão
        se end \neq \lambda então i \leftarrow end :
        senão
5
             i \leftarrow 1; j \leftarrow h(x)
 6
             enquanto i \leq m faça
 7
                   se t[i].estado = ocupado então
 8
                       j \leftarrow (j+1) \mod m
 9
                        i \leftarrow i + 1
10
                   senão
11
                      i \leftarrow m + 2
12
              se i = m + 1 então "overflow"; pare ;
13
```

temp ← T[/].pont %fusão de listas

 $T[i].pont \leftarrow T[h(x)].pont$

 $T[h(x)].pont \leftarrow temp$

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadea-

Encadeamer

Encadeame

Em duna Zana

Lili dudo Zorido

```
Algoritmo 47: Inserir(x, end, a)
 Buscar(x, end, a)
 2 se a = 1 então "Inserção inválida: chave já existe";
3 senão
        se end \neq \lambda então i \leftarrow end :
        senão
 5
              i \leftarrow 1; j \leftarrow h(x)
 6
              enquanto i \leq m faça
 7
                   se t[i].estado = ocupado então
 8
                        j \leftarrow (j+1) \mod m
 9
                        i \leftarrow i + 1
10
                   senão
11
                        i \leftarrow m + 2
12
              se i = m + 1 então "overflow"; pare ;
13
              temp ← T[/].pont %fusão de listas
14
              T[i].pont \leftarrow T[h(x)].pont
15
              T[h(x)].pont \leftarrow temp
16
17
         T[i].chave \leftarrow x %inserção da chave
         T[i].estado ← ocupado
18
```

Bibliografia

EDA - Aula 6

Aula Passad:

Tratamento d Colisão por Encadeamento

Encadeamen Exterior

Interior

Em duas Zonas

SZWARCFITER, Jayme; MARKENZON, Lilian. Estruturas de Dados e Seus Algoritmos.3a edição. LTC, ano 2010. (ISBN 9788521617501).

EDA - Aula 6

Prof. Eurinardo

Aula Passada

Tratamento de Colisão por Encadeamento

Encadeamento

EXTRIBUT

IIIterioi

Em uma única zona

Obrigado!