

MATA54 - Estruturas de Dados e Algoritmos II

Hashing - Encadeamento Explícito

Flávio Assis

Versão gerada a partir de slides do Prof. George Lima

IC - Instituto de Computação

Salvador, agosto de 2021

Motivação: busca em $O(1)$

Mapeamento entre chave-primária e endereço

- ▶ Podemos usar uma função $h(k)$ com domínio no espaço de chaves e contra-domínio no espaço de endereços $0..m - 1$
- ▶ Usa-se $h(k)$ para se definir onde k ficará armazenada no arquivo

$$h(k) \rightarrow \{0, 1, \dots, m - 1\}$$

End	Chave	Info
0:	k_1	r_1
1:	-	-
2:	k_2	r_2
...
$m - 1$:	k_n	r_n

Motivação: busca em $O(1)$

Mapeamento entre chave-primária e endereço

- Podemos usar uma função $h(k)$ com domínio no espaço de chaves e contra-domínio no espaço de endereços $0..m - 1$
- Usa-se $h(k)$ para se definir onde k ficará armazenada no arquivo

$$h(k) \rightarrow \{0, 1, \dots, m - 1\}$$

End	Chave	Info
0:	k_1	r_1
1:	-	-
2:	k_2	r_2
...
$m - 1$:	k_n	r_n

- Dada uma chave k , $h(k)$ retorna uma posição onde a chave k está ou a partir da qual se iniciará a busca pela chave

Motivação: busca em $O(1)$

Mapeamento entre chave-primária e endereço

- ▶ Podemos usar uma função $h(k)$ com domínio no espaço de chaves e contra-domínio no espaço de endereços $0..m - 1$
- ▶ Usa-se $h(k)$ para se definir onde k ficará armazenada no arquivo

$$h(k) \rightarrow \{0, 1, \dots, m - 1\}$$

End	Chave	Info
0:	k_1	r_1
1:	-	-
2:	k_2	r_2
...
$m - 1$:	k_n	r_n

- ▶ Dada uma chave k , $h(k)$ retorna uma posição onde a chave k está ou a partir da qual se iniciará a busca pela chave
- ▶ A função $h(k)$ é chamada de **função de hashing**, pois deve **espalhar** as chaves no espaço de endereços

Motivação: busca em $O(1)$

Mapeamento entre chave-primária e endereço

- ▶ E se usarmos uma função **injetora** $h(k)$ que retorna o índice do registro com chave k ?
- ▶ Acesso ao registro seria $O(1)$ no pior caso!

End	Chave	Info
0:	k_1	r_1
1:	-	-
2:	k_2	r_2
...
$m - 1$:	k_n	r_n

Motivação: busca em $O(1)$

Mapeamento entre chave-primária e endereço

- ▶ E se usarmos uma função **injetora** $h(k)$ que retorna o índice do registro com chave k ?
- ▶ Acesso ao registro seria $O(1)$ no pior caso!

End	Chave	Info
0:	k_1	r_1
1:	-	-
2:	k_2	r_2
...
$m - 1$:	k_n	r_n

Qual o problema desta abordagem?

Motivação: busca em $O(1)$

Mapeamento entre chave-primária e endereço

- ▶ E se usarmos uma função **bijetora** $h(k)$ que mapeasse n chaves no espaço de endereços $0..n-1$?
- ▶ Acesso ao registro seria $O(1)$ no pior caso e não haveria perda de espaço!

End	Chave	Info
0:	k_1	r_1
1:	k_2	r_2
2:	k_3	r_3
...
$n-1$:	k_n	r_n

Motivação: busca em $O(1)$

Mapeamento entre chave-primária e endereço

- ▶ E se usarmos uma função **bijetora** $h(k)$ que mapeasse n chaves no espaço de endereços $0..n-1$?
- ▶ Acesso ao registro seria $O(1)$ no pior caso e não haveria perda de espaço!

End	Chave	Info
0:	k_1	r_1
1:	k_2	r_2
2:	k_3	r_3
...
$n-1$:	k_n	r_n

Qual o problema desta abordagem?

É difícil obter tais funções em geral. Pode-se obter para alguns conjuntos de chaves fixos.

Mapeamento entre chave-primária e endereço

- ▶ **Função hashing**: função **sobrejetora** de chave para endereço.
 - ▶ **Hipótese: hashing uniforme simples**
Qualquer chave é mapeada em qualquer um dos m endereços com igual probabilidade.

Exemplo:

$$h(k) = k \bmod m$$

Mapeamento entre chave-primária e endereço

- ▶ **Função hashing**: função **sobrejetora** de chave para endereço.
 - ▶ **Hipótese: hashing uniforme simples**
Qualquer chave é mapeada em qualquer um dos m endereços com igual probabilidade.

Exemplo:

$$h(k) = k \bmod m$$

- ▶ **Colisão**: Chaves distintas podem ser mapeadas no mesmo endereço.
Abordagens via encadeamento:
 - ▶ **Explícito**: usa ponteiros
 - ▶ **Endereçamento aberto**: não usa ponteiros

Encadeamento explícito e alocação dinâmica

- ▶ Suponhamos que haja espaço além da tabela para armazenamento
- ▶ **Listas crescem além do tamanho da tabela hashing**
- ▶ Ex.: 27, 18, 29, 43, 77, 16 e 40; $m = 11$; e $h(k) = k \bmod m$.

End Listas de registros

0:	\emptyset
1:	\emptyset
2:	\emptyset
3:	\emptyset
4:	\emptyset
5:	\emptyset
6:	\emptyset
7:	\emptyset
8:	\emptyset
9:	\emptyset
10:	\emptyset

Encadeamento explícito e alocação dinâmica

- Suponhamos que haja espaço além da tabela para armazenamento
- **Listas crescem além do tamanho da tabela hashing**
- Ex.: 27, 18, 29, 43, 77, 16 e 40; $m = 11$; e $h(k) = k \bmod m$.

End Listas de registros

0:	∅
1:	∅
2:	∅
3:	∅
4:	∅
5:	∅
6:	∅
7:	∅
8:	∅
9:	∅
10:	∅

End Listas de registros

0:	∅
1:	∅
2:	∅
3:	∅
4:	∅
5:	27
6:	∅
7:	∅
8:	∅
9:	∅
10:	∅

→ ∅

Encadeamento explícito e alocação dinâmica

- Suponhamos que haja espaço além da tabela para armazenamento
- **Listas crescem além do tamanho da tabela hashing**
- Ex.: 27, 18, 29, 43, 77, 16 e 40; $m = 11$; e $h(k) = k \bmod m$.

End Listas de registros

0:	<div>∅</div>
1:	<div>∅</div>
2:	<div>∅</div>
3:	<div>∅</div>
4:	<div>∅</div>
5:	<div>∅</div>
6:	<div>∅</div>
7:	<div>∅</div>
8:	<div>∅</div>
9:	<div>∅</div>
10:	<div>∅</div>

End Listas de registros

0:	<div>∅</div>
1:	<div>∅</div>
2:	<div>∅</div>
3:	<div>∅</div>
4:	<div>∅</div>
5:	<div>27</div>
6:	<div>∅</div>
7:	<div>∅</div>
8:	<div>∅</div>
9:	<div>∅</div>
10:	<div>∅</div>

→ ∅

End Listas de registros

0:	<div>∅</div>
1:	<div>∅</div>
2:	<div>∅</div>
3:	<div>∅</div>
4:	<div>∅</div>
5:	<div>27</div>
6:	<div>∅</div>
7:	<div>18</div>
8:	<div>∅</div>
9:	<div>∅</div>
10:	<div>∅</div>

→ ∅

Encadeamento explícito e alocação dinâmica

- ▶ Suponhamos que haja espaço além da tabela para armazenamento
- ▶ **Listas crescem além do tamanho da tabela hashing**
- ▶ Ex.: 27, 18, 29, 43, 77, 16 e 40; $m = 11$; e $h(k) = k \bmod m$.

End	Listas de registros		
0:	<table><tr><td>\emptyset</td></tr></table>	\emptyset	
\emptyset			
1:	<table><tr><td>\emptyset</td></tr></table>	\emptyset	
\emptyset			
2:	<table><tr><td>\emptyset</td></tr></table>	\emptyset	
\emptyset			
3:	<table><tr><td>\emptyset</td></tr></table>	\emptyset	
\emptyset			
4:	<table><tr><td>\emptyset</td></tr></table>	\emptyset	
\emptyset			
5:	<table><tr><td>27</td></tr></table>	27	$\rightarrow \emptyset$
27			
6:	<table><tr><td>\emptyset</td></tr></table>	\emptyset	
\emptyset			
7:	<table><tr><td>18</td></tr></table>	18	$\rightarrow 29 \rightarrow \emptyset$
18			
8:	<table><tr><td>\emptyset</td></tr></table>	\emptyset	
\emptyset			
9:	<table><tr><td>\emptyset</td></tr></table>	\emptyset	
\emptyset			
10:	<table><tr><td>\emptyset</td></tr></table>	\emptyset	
\emptyset			

Encadeamento explícito e alocação dinâmica

- Suponhamos que haja espaço além da tabela para armazenamento
- **Listas crescem além do tamanho da tabela hashing**
- Ex.: 27, 18, 29, 43, 77, 16 e 40; $m = 11$; e $h(k) = k \bmod m$.

End Listas de registros

0:	<div>∅</div>	
1:	<div>∅</div>	
2:	<div>∅</div>	
3:	<div>∅</div>	
4:	<div>∅</div>	
5:	<div>27</div>	→ ∅
6:	<div>∅</div>	
7:	<div>18</div>	→ 29 → ∅
8:	<div>∅</div>	
9:	<div>∅</div>	
10:	<div>∅</div>	

End Listas de registros

0:	<div>∅</div>	
1:	<div>∅</div>	
2:	<div>∅</div>	
3:	<div>∅</div>	
4:	<div>∅</div>	
5:	<div>27</div>	→ ∅
6:	<div>∅</div>	
7:	<div>18</div>	→ 29 → ∅
8:	<div>∅</div>	
9:	<div>∅</div>	
10:	<div>43</div>	→ ∅

Encadeamento explícito e alocação dinâmica

- Suponhamos que haja espaço além da tabela para armazenamento
- **Listas crescem além do tamanho da tabela hashing**
- Ex.: 27, 18, 29, 43, 77, 16 e 40; $m = 11$; e $h(k) = k \bmod m$.

End	Listas de registros	
0:	<table><tr><td>∅</td></tr></table>	∅
∅		
1:	<table><tr><td>∅</td></tr></table>	∅
∅		
2:	<table><tr><td>∅</td></tr></table>	∅
∅		
3:	<table><tr><td>∅</td></tr></table>	∅
∅		
4:	<table><tr><td>∅</td></tr></table>	∅
∅		
5:	<table><tr><td>27</td></tr></table> → ∅	27
27		
6:	<table><tr><td>∅</td></tr></table>	∅
∅		
7:	<table><tr><td>18</td></tr></table> → 29 → ∅	18
18		
8:	<table><tr><td>∅</td></tr></table>	∅
∅		
9:	<table><tr><td>∅</td></tr></table>	∅
∅		
10:	<table><tr><td>∅</td></tr></table>	∅
∅		

End	Listas de registros		
0:	<table><tr><td>∅</td></tr></table>	∅	
∅			
1:	<table><tr><td>∅</td></tr></table>	∅	
∅			
2:	<table><tr><td>∅</td></tr></table>	∅	
∅			
3:	<table><tr><td>∅</td></tr></table>	∅	
∅			
4:	<table><tr><td>∅</td></tr></table>	∅	
∅			
5:	<table><tr><td>27</td></tr></table>	27	→ ∅
27			
6:	<table><tr><td>∅</td></tr></table>	∅	
∅			
7:	<table><tr><td>18</td></tr></table>	18	→ 29 → ∅
18			
8:	<table><tr><td>∅</td></tr></table>	∅	
∅			
9:	<table><tr><td>∅</td></tr></table>	∅	
∅			
10:	<table><tr><td>43</td></tr></table>	43	→ ∅
43			

End	Listas de registros	
0:	<table><tr><td>77</td></tr></table> → ∅	77
77		
1:	<table><tr><td>∅</td></tr></table>	∅
∅		
2:	<table><tr><td>∅</td></tr></table>	∅
∅		
3:	<table><tr><td>∅</td></tr></table>	∅
∅		
4:	<table><tr><td>∅</td></tr></table>	∅
∅		
5:	<table><tr><td>27</td></tr></table> → ∅	27
27		
6:	<table><tr><td>∅</td></tr></table>	∅
∅		
7:	<table><tr><td>18</td></tr></table> → 29 → ∅	18
18		
8:	<table><tr><td>∅</td></tr></table>	∅
∅		
9:	<table><tr><td>∅</td></tr></table>	∅
∅		
10:	<table><tr><td>43</td></tr></table> → ∅	43
43		

Encadeamento explícito e alocação dinâmica

- Suponhamos que haja espaço além da tabela para armazenamento
- **Listas crescem além do tamanho da tabela hashing**
- Ex.: 27, 18, 29, 43, 77, 16 e 40; $m = 11$; e $h(k) = k \bmod m$.

End	Listas de registros	
0:	<table><tr><td>77</td></tr></table> → ∅	77
77		
1:	<table><tr><td>∅</td></tr></table>	∅
∅		
2:	<table><tr><td>13</td></tr></table> → ∅	13
13		
3:	<table><tr><td>∅</td></tr></table>	∅
∅		
4:	<table><tr><td>∅</td></tr></table>	∅
∅		
5:	<table><tr><td>27</td></tr></table> → 16 → ∅	27
27		
6:	<table><tr><td>∅</td></tr></table>	∅
∅		
7:	<table><tr><td>18</td></tr></table> → 29 → ∅	18
18		
8:	<table><tr><td>∅</td></tr></table>	∅
∅		
9:	<table><tr><td>∅</td></tr></table>	∅
∅		
10:	<table><tr><td>43</td></tr></table> → ∅	43
43		

Encadeamento explícito e alocação dinâmica

- Suponhamos que haja espaço além da tabela para armazenamento
- **Listas crescem além do tamanho da tabela hashing**
- Ex.: 27, 18, 29, 43, 77, 16 e 40; $m = 11$; e $h(k) = k \bmod m$.

End	Listas de registros	
0:	<table><tr><td>77</td></tr></table> → ∅	77
77		
1:	<table><tr><td>∅</td></tr></table>	∅
∅		
2:	<table><tr><td>13</td></tr></table> → ∅	13
13		
3:	<table><tr><td>∅</td></tr></table>	∅
∅		
4:	<table><tr><td>∅</td></tr></table>	∅
∅		
5:	<table><tr><td>27</td></tr></table> → 16 → ∅	27
27		
6:	<table><tr><td>∅</td></tr></table>	∅
∅		
7:	<table><tr><td>18</td></tr></table> → 29 → ∅	18
18		
8:	<table><tr><td>∅</td></tr></table>	∅
∅		
9:	<table><tr><td>∅</td></tr></table>	∅
∅		
10:	<table><tr><td>43</td></tr></table> → ∅	43
43		

End	Listas de registros		
0:	<table><tr><td>77</td></tr></table>	77	→ ∅
77			
1:	<table><tr><td>∅</td></tr></table>	∅	
∅			
2:	<table><tr><td>13</td></tr></table>	13	→ ∅
13			
3:	<table><tr><td>∅</td></tr></table>	∅	
∅			
4:	<table><tr><td>∅</td></tr></table>	∅	
∅			
5:	<table><tr><td>27</td></tr></table>	27	→ 16 → ∅
27			
6:	<table><tr><td>∅</td></tr></table>	∅	
∅			
7:	<table><tr><td>18</td></tr></table>	18	→ 29 → 40 → ∅
18			
8:	<table><tr><td>∅</td></tr></table>	∅	
∅			
9:	<table><tr><td>∅</td></tr></table>	∅	
∅			
10:	<table><tr><td>43</td></tr></table>	43	→ ∅
43			

Encadeamento explícito e alocação dinâmica

- ▶ Suponhamos que haja espaço além da tabela para armazenamento
- ▶ **Listas crescem além do tamanho da tabela hashing**
- ▶ Ex.: 27, 18, 29, 43, 77, 16 e 40; $m = 11$; e $h(k) = k \bmod m$.

End	Listas de registros	
0:	<table><tr><td>77</td></tr></table> → ∅	77
77		
1:	<table><tr><td>∅</td></tr></table>	∅
∅		
2:	<table><tr><td>13</td></tr></table> → ∅	13
13		
3:	<table><tr><td>∅</td></tr></table>	∅
∅		
4:	<table><tr><td>∅</td></tr></table>	∅
∅		
5:	<table><tr><td>27</td></tr></table> → 16 → ∅	27
27		
6:	<table><tr><td>∅</td></tr></table>	∅
∅		
7:	<table><tr><td>18</td></tr></table> → 29 → ∅	18
18		
8:	<table><tr><td>∅</td></tr></table>	∅
∅		
9:	<table><tr><td>∅</td></tr></table>	∅
∅		
10:	<table><tr><td>43</td></tr></table> → ∅	43
43		

End	Listas de registros		
0:	<table><tr><td>77</td></tr></table>	77	→ ∅
77			
1:	<table><tr><td>∅</td></tr></table>	∅	
∅			
2:	<table><tr><td>13</td></tr></table>	13	→ ∅
13			
3:	<table><tr><td>∅</td></tr></table>	∅	
∅			
4:	<table><tr><td>∅</td></tr></table>	∅	
∅			
5:	<table><tr><td>27</td></tr></table>	27	→ 16 → ∅
27			
6:	<table><tr><td>∅</td></tr></table>	∅	
∅			
7:	<table><tr><td>18</td></tr></table>	18	→ 29 → 40 → ∅
18			
8:	<table><tr><td>∅</td></tr></table>	∅	
∅			
9:	<table><tr><td>∅</td></tr></table>	∅	
∅			
10:	<table><tr><td>43</td></tr></table>	43	→ ∅
43			

Com **fator de carga** $\alpha = \frac{n}{m}$, complexidade (caso médio) é $O(1 + \alpha)$ sob hipótese de hashing uniforme simples

Vantagens e desvantagens?

Encadeamento explícito e alocação estática

- ▶ Listas encadeadas confinadas no espaço de endereçamento previamente alocado
- ▶ Fator de carga $0 \leq \alpha \leq 1$
- ▶ Ex.: 27, 18, 29, 43, 77, 16 e 40; $m = 11$; e $h(k) = k \bmod m$.

End	Reg.	Pt
0:		\emptyset
1:		\emptyset
2:		\emptyset
3:		\emptyset
4:		\emptyset
5:		\emptyset
6:		\emptyset
7:		\emptyset
8:		\emptyset
9:		\emptyset
→ 10:		\emptyset

Encadeamento explícito e alocação estática

- ▶ Listas encadeadas confinadas no espaço de endereçamento previamente alocado
- ▶ Fator de carga $0 \leq \alpha \leq 1$
- ▶ Ex.: 27, 18, 29, 43, 77, 16 e 40; $m = 11$; e $h(k) = k \bmod m$.

End	Reg.	Pt
0:		\emptyset
1:		\emptyset
2:		\emptyset
3:		\emptyset
4:		\emptyset
5:		\emptyset
6:		\emptyset
7:		\emptyset
8:		\emptyset
9:		\emptyset
→ 10:		\emptyset

End	Reg.	Pt
0:		\emptyset
1:		\emptyset
2:		\emptyset
3:		\emptyset
4:		\emptyset
5:	27	\emptyset
6:		\emptyset
7:		\emptyset
8:		\emptyset
9:		\emptyset
→ 10:		\emptyset

Encadeamento explícito e alocação estática

- ▶ Listas encadeadas confinadas no espaço de endereçamento previamente alocado
- ▶ Fator de carga $0 \leq \alpha \leq 1$
- ▶ Ex.: 27, 18, 29, 43, 77, 16 e 40; $m = 11$; e $h(k) = k \bmod m$.

End	Reg.	Pt
0:		\emptyset
1:		\emptyset
2:		\emptyset
3:		\emptyset
4:		\emptyset
5:		\emptyset
6:		\emptyset
7:		\emptyset
8:		\emptyset
9:		\emptyset
→ 10:		\emptyset

End	Reg.	Pt
0:		\emptyset
1:		\emptyset
2:		\emptyset
3:		\emptyset
4:		\emptyset
5:	27	\emptyset
6:		\emptyset
7:		\emptyset
8:		\emptyset
9:		\emptyset
→ 10:		\emptyset

End	Reg.	Pt
0:		\emptyset
1:		\emptyset
2:		\emptyset
3:		\emptyset
4:		\emptyset
5:	27	\emptyset
6:		\emptyset
7:	18	\emptyset
8:		\emptyset
9:		\emptyset
→ 10:		\emptyset

Encadeamento explícito e alocação estática

- ▶ Listas encadeadas confinadas no espaço de endereçamento previamente alocado
- ▶ Fator de carga $0 \leq \alpha \leq 1$
- ▶ Ex.: 27, 18, 29, 43, 77, 16 e 40; $m = 11$; e $h(k) = k \bmod m$.

End	Reg.	Pt
0:		\emptyset
1:		\emptyset
2:		\emptyset
3:		\emptyset
4:		\emptyset
5:	27	\emptyset
6:		\emptyset
7:	18	10
8:		\emptyset
→ 9:		\emptyset
10:	29	\emptyset

Encadeamento explícito e alocação estática

- ▶ Listas encadeadas confinadas no espaço de endereçamento previamente alocado
- ▶ Fator de carga $0 \leq \alpha \leq 1$
- ▶ Ex.: 27, 18, 29, 43, 77, 16 e 40; $m = 11$; e $h(k) = k \bmod m$.

End	Reg.	Pt
0:		\emptyset
1:		\emptyset
2:		\emptyset
3:		\emptyset
4:		\emptyset
5:	27	\emptyset
6:		\emptyset
7:	18	10
8:		\emptyset
→ 9:		\emptyset
10:	29	\emptyset

End	Reg.	Pt
0:		\emptyset
1:		\emptyset
2:		\emptyset
3:		\emptyset
4:		\emptyset
5:	27	\emptyset
6:		\emptyset
7:	18	10
→ 8:		\emptyset
9:	43	\emptyset
10:	29	9

Encadeamento explícito e alocação estática

- ▶ Listas encadeadas confinadas no espaço de endereçamento previamente alocado
- ▶ Fator de carga $0 \leq \alpha \leq 1$
- ▶ Ex.: 27, 18, 29, 43, 77, 16 e 40; $m = 11$; e $h(k) = k \bmod m$.

End	Reg.	Pt
0:		\emptyset
1:		\emptyset
2:		\emptyset
3:		\emptyset
4:		\emptyset
5:	27	\emptyset
6:		\emptyset
7:	18	10
8:		\emptyset
→ 9:		\emptyset
10:	29	\emptyset

End	Reg.	Pt
0:		\emptyset
1:		\emptyset
2:		\emptyset
3:		\emptyset
4:		\emptyset
5:	27	\emptyset
6:		\emptyset
7:	18	10
→ 8:		\emptyset
9:	43	\emptyset
10:	29	9

End	Reg.	Pt
0:	77	\emptyset
1:		\emptyset
2:		\emptyset
3:		\emptyset
4:		\emptyset
5:	27	\emptyset
6:		\emptyset
7:	18	10
→ 8:		\emptyset
9:	43	\emptyset
10:	29	9

Encadeamento explícito e alocação estática

- ▶ Listas encadeadas confinadas no espaço de endereçamento previamente alocado
- ▶ Fator de carga $0 \leq \alpha \leq 1$
- ▶ Ex.: 27, 18, 29, 43, 77, 16 e 40; $m = 11$; e $h(k) = k \bmod m$.

End	Reg.	Pt
0:	77	\emptyset
1:		\emptyset
2:		\emptyset
3:		\emptyset
4:		\emptyset
5:	27	8
→ 6:		\emptyset
7:	18	10
8:	16	\emptyset
9:	43	\emptyset
10:	29	9

Encadeamento explícito e alocação estática

- ▶ Listas encadeadas confinadas no espaço de endereçamento previamente alocado
- ▶ Fator de carga $0 \leq \alpha \leq 1$
- ▶ Ex.: 27, 18, 29, 43, 77, 16 e 40; $m = 11$; e $h(k) = k \bmod m$.

End	Reg.	Pt
0:	77	\emptyset
1:		\emptyset
2:		\emptyset
3:		\emptyset
4:		\emptyset
5:	27	8
→ 6:		\emptyset
7:	18	10
8:	16	\emptyset
9:	43	\emptyset
10:	29	9

End	Reg.	Pt
0:	77	\emptyset
1:		\emptyset
2:		\emptyset
3:		\emptyset
→ 4:		\emptyset
5:	27	8
6:	40	\emptyset
7:	18	10
8:	16	\emptyset
9:	43	6
10:	29	9

Encadeamento explícito e alocação estática

- ▶ Listas encadeadas confinadas no espaço de endereçamento previamente alocado
- ▶ Fator de carga $0 \leq \alpha \leq 1$
- ▶ Ex.: 27, 18, 29, 43, 77, 16 e 40; $m = 11$; e $h(k) = k \bmod m$.

End	Reg.	Pt
0:	77	\emptyset
1:		\emptyset
2:		\emptyset
3:		\emptyset
4:		\emptyset
5:	27	8
→ 6:		\emptyset
7:	18	10
8:	16	\emptyset
9:	43	\emptyset
10:	29	9

End	Reg.	Pt
0:	77	\emptyset
1:		\emptyset
2:		\emptyset
3:		\emptyset
→ 4:		\emptyset
5:	27	8
6:	40	\emptyset
7:	18	10
8:	16	\emptyset
9:	43	6
10:	29	9

Vantagens e desvantagens?

Exercício

Qual seria o estado final de um arquivo usando hash com endereçamento explícito com $m = 11$ e $h(k) = k \bmod m$, após a inserção da seguinte sequência de chaves:

16, 19, 38, 10, 31, 11, 27, 13

Exercício

Qual seria o estado final de um arquivo usando hash com endereçamento explícito com $m = 11$ e $h(k) = k \bmod m$, após a inserção da seguinte sequência de chaves:

16, 19, 38, 10, 31, 11, 27, 13

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:		∅
5:		∅
6:		∅
7:		∅
8:		∅
9:		∅
→ 10:		∅

Exercício

Qual seria o estado final de um arquivo usando hash com endereçamento explícito com $m = 11$ e $h(k) = k \bmod m$, após a inserção da seguinte sequência de chaves:

16, 19, 38, 10, 31, 11, 27, 13

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:		∅
5:		∅
6:		∅
7:		∅
8:		∅
9:		∅
→ 10:		∅

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:		∅
5:	16	∅
6:		∅
7:		∅
8:		∅
9:		∅
→ 10:		∅

Exercício

Qual seria o estado final de um arquivo usando hash com endereçamento explícito com $m = 11$ e $h(k) = k \bmod m$, após a inserção da seguinte sequência de chaves:

16, 19, 38, 10, 31, 11, 27, 13

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:		∅
5:		∅
6:		∅
7:		∅
8:		∅
9:		∅
→ 10:		∅

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:		∅
5:	16	∅
6:		∅
7:		∅
8:		∅
9:		∅
→ 10:		∅

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:		∅
5:	16	∅
6:		∅
7:		∅
8:	19	∅
9:		∅
→ 10:		∅

Exercício

Qual seria o estado final de um arquivo usando hash com endereçamento explícito com $m = 11$ e $h(k) = k \bmod m$, após a inserção da seguinte sequência de chaves:

16, 19, 38, 10, 31, 11, 27, 13

End	Reg.	Pt
0:		\emptyset
1:		\emptyset
2:		\emptyset
3:		\emptyset
4:		\emptyset
5:	16	10
6:		\emptyset
7:		\emptyset
8:	19	\emptyset
→ 9:		\emptyset
10:	38	\emptyset

Exercício

Qual seria o estado final de um arquivo usando hash com endereçamento explícito com $m = 11$ e $h(k) = k \bmod m$, após a inserção da seguinte sequência de chaves:

16, 19, 38, 10, 31, 11, 27, 13

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:		∅
5:	16	10
6:		∅
7:		∅
8:	19	∅
→ 9:		∅
10:	38	∅

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:		∅
5:	16	10
6:		∅
→ 7:		∅
8:	19	∅
9:	10	∅
10:	38	9

Exercício

Qual seria o estado final de um arquivo usando hash com endereçamento explícito com $m = 11$ e $h(k) = k \bmod m$, após a inserção da seguinte sequência de chaves:

16, 19, 38, 10, 31, 11, 27, 13

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:		∅
5:	16	10
6:		∅
7:		∅
8:	19	∅
→ 9:		∅
10:	38	∅

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:		∅
5:	16	10
6:		∅
→ 7:		∅
8:	19	∅
9:	10	∅
10:	38	9

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:		∅
5:	16	10
→ 6:		∅
7:	31	∅
8:	19	∅
9:	10	7
10:	38	9

Exercício

Qual seria o estado final de um arquivo usando hash com endereçamento explícito com $m = 11$ e $h(k) = k \bmod m$, após a inserção da seguinte sequência de chaves:

16, 19, 38, 10, 31, 11, 27, 13

End	Reg.	Pt
0:	11	\emptyset
1:		\emptyset
2:		\emptyset
3:		\emptyset
4:		\emptyset
5:	16	10
→ 6:		\emptyset
7:	31	\emptyset
8:	19	\emptyset
9:	10	7
10:	38	9

Exercício

Qual seria o estado final de um arquivo usando hash com endereçamento explícito com $m = 11$ e $h(k) = k \bmod m$, após a inserção da seguinte sequência de chaves:

16, 19, 38, 10, 31, 11, 27, 13

End	Reg.	Pt
0:	11	\emptyset
1:		\emptyset
2:		\emptyset
3:		\emptyset
4:		\emptyset
5:	16	10
→ 6:		\emptyset
7:	31	\emptyset
8:	19	\emptyset
9:	10	7
10:	38	9

End	Reg.	Pt
0:	11	\emptyset
1:		\emptyset
2:		\emptyset
3:		\emptyset
→ 4:		\emptyset
5:	16	10
6:	27	\emptyset
7:	31	6
8:	19	\emptyset
9:	10	7
10:	38	9

Exercício

Qual seria o estado final de um arquivo usando hash com endereçamento explícito com $m = 11$ e $h(k) = k \bmod m$, após a inserção da seguinte sequência de chaves:

16, 19, 38, 10, 31, 11, 27, 13

End	Reg.	Pt
0:	11	∅
1:		∅
2:		∅
3:		∅
4:		∅
5:	16	10
→ 6:		∅
7:	31	∅
8:	19	∅
9:	10	7
10:	38	9

End	Reg.	Pt
0:	11	∅
1:		∅
2:		∅
3:		∅
→ 4:		∅
5:	16	10
6:	27	∅
7:	31	6
8:	19	∅
9:	10	7
10:	38	9

End	Reg.	Pt
0:	11	∅
1:		∅
2:	13	∅
3:		∅
→ 4:		∅
5:	16	10
6:	27	∅
7:	31	6
8:	19	∅
9:	10	7
10:	38	9

Exercício

Qual seria o estado final de um arquivo usando hash com endereçamento explícito com $m = 11$ e $h(k) = k \bmod m$, após a inserção da seguinte sequência de chaves:

16, 19, 38, 10, 31, 11, 27, 13

End	Reg.	Pt
0:	11	\emptyset
1:		\emptyset
2:		\emptyset
3:		\emptyset
4:		\emptyset
5:	16	10
→ 6:		\emptyset
7:	31	\emptyset
8:	19	\emptyset
9:	10	7
10:	38	9

End	Reg.	Pt
0:	11	\emptyset
1:		\emptyset
2:		\emptyset
3:		\emptyset
→ 4:		\emptyset
5:	16	10
6:	27	\emptyset
7:	31	6
8:	19	\emptyset
9:	10	7
10:	38	9

End	Reg.	Pt
0:	11	\emptyset
1:		\emptyset
2:	13	\emptyset
3:		\emptyset
→ 4:		\emptyset
5:	16	10
6:	27	\emptyset
7:	31	6
8:	19	\emptyset
9:	10	7
10:	38	9

Média de acessos: $\frac{1+1+2+2+2+1+5+1}{8} = 1,9$

Encadeamento Explícito

- ▶ O esquema anterior resulta em **junção de cadeias**: para se encontrar uma chave k pode-se passar por registros cujas chaves possuem diferentes valores de $h(k)$

Encadeamento Explícito

- ▶ O esquema anterior resulta em **junção de cadeias**: para se encontrar uma chave k pode-se passar por registros cujas chaves possuem diferentes valores de $h(k)$
- ▶ Como manter o encadeamento explícito **sem** junção de cadeias?

Encadeamento Explícito

- ▶ O esquema anterior resulta em **junção de cadeias**: para se encontrar uma chave k pode-se passar por registros cujas chaves possuem diferentes valores de $h(k)$
- ▶ Como manter o encadeamento explícito **sem** junção de cadeias?
- ▶ Em particular, como fazer inserção, consulta e remoção de registros?

Encadeamento Explícito sem Junção de Cadeias

Ao se inserir um registro com chave k :

1. se a posição $h(k)$ estiver livre, insere-se o registro nessa posição

Encadeamento Explícito sem Junção de Cadeias

Ao se inserir um registro com chave k :

1. se a posição $h(k)$ estiver livre, insere-se o registro nessa posição
2. se a posição $h(k)$ estiver ocupada por um registro com chave com mesmo endereço original de k , insere-se o novo registro na cadeia que se inicia nesta posição

Encadeamento Explícito sem Junção de Cadeias

Ao se inserir um registro com chave k :

1. se a posição $h(k)$ estiver livre, insere-se o registro nessa posição
2. se a posição $h(k)$ estiver ocupada por um registro com chave com mesmo endereço original de k , insere-se o novo registro na cadeia que se inicia nesta posição
3. senão, realoca-se o registro que estava na posição $h(k)$ para uma posição vazia e insere-se o novo registro em $h(k)$

Encadeamento Explícito sem Junção de Cadeias

Ao se inserir um registro com chave k :

1. se a posição $h(k)$ estiver livre, insere-se o registro nessa posição
2. se a posição $h(k)$ estiver ocupada por um registro com chave com mesmo endereço original de k , insere-se o novo registro na cadeia que se inicia nesta posição
3. senão, realoca-se o registro que estava na posição $h(k)$ para uma posição vazia e insere-se o novo registro em $h(k)$

Exemplo: 15, 28, 19, 26, 4, 30, 41, 18 - $h(k) = k \bmod m$, $m = 11$

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:		∅
5:		∅
6:		∅
7:		∅
8:		∅
9:		∅
→ 10:		∅

Encadeamento Explícito sem Junção de Cadeias

Ao se inserir um registro com chave k :

1. se a posição $h(k)$ estiver livre, insere-se o registro nessa posição
2. se a posição $h(k)$ estiver ocupada por um registro com chave com mesmo endereço original de k , insere-se o novo registro na cadeia que se inicia nesta posição
3. senão, realoca-se o registro que estava na posição $h(k)$ para uma posição vazia e insere-se o novo registro em $h(k)$

Exemplo: 15, 28, 19, 26, 4, 30, 41, 18 - $h(k) = k \bmod m$, $m = 11$

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:		∅
5:		∅
6:		∅
7:		∅
8:		∅
9:		∅
→ 10:		∅

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:	15	∅
5:		∅
6:		∅
7:		∅
8:		∅
9:		∅
→ 10:		∅

Encadeamento Explícito sem Junção de Cadeias

Ao se inserir um registro com chave k :

1. se a posição $h(k)$ estiver livre, insere-se o registro nessa posição
2. se a posição $h(k)$ estiver ocupada por um registro com chave com mesmo endereço original de k , insere-se o novo registro na cadeia que se inicia nesta posição
3. senão, realoca-se o registro que estava na posição $h(k)$ para uma posição vazia e insere-se o novo registro em $h(k)$

Exemplo: 15, 28, 19, 26, 4, 30, 41, 18 - $h(k) = k \bmod m$, $m = 11$

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:		∅
5:		∅
6:		∅
7:		∅
8:		∅
9:		∅
→ 10:		∅

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:	15	∅
5:		∅
6:		∅
7:		∅
8:		∅
9:		∅
→ 10:		∅

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:	15	∅
5:		∅
6:	28	∅
7:		∅
8:		∅
9:		∅
→ 10:		∅

Encadeamento Explícito sem Junção de Cadeias

Ao se inserir um registro com chave k :

1. se a posição $h(k)$ estiver livre, insere-se o registro nessa posição
2. se a posição $h(k)$ estiver ocupada por um registro com chave com mesmo endereço original de k , insere-se o novo registro na cadeia que se inicia nesta posição
3. senão, realoca-se o registro que estava na posição $h(k)$ para uma posição vazia e insere-se o novo registro em $h(k)$

Exemplo: 15, 28, 19, 26, 4, 30, 41, 18 - $h(k) = k \bmod m$, $m = 11$

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:	15	∅
5:		∅
6:	28	∅
7:		∅
8:	19	∅
9:		∅
→ 10:		∅

Encadeamento Explícito sem Junção de Cadeias

Ao se inserir um registro com chave k :

1. se a posição $h(k)$ estiver livre, insere-se o registro nessa posição
2. se a posição $h(k)$ estiver ocupada por um registro com chave com mesmo endereço original de k , insere-se o novo registro na cadeia que se inicia nesta posição
3. senão, realoca-se o registro que estava na posição $h(k)$ para uma posição vazia e insere-se o novo registro em $h(k)$

Exemplo: 15, 28, 19, 26, 4, 30, 41, 18 - $h(k) = k \bmod m$, $m = 11$

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:	15	∅
5:		∅
6:	28	∅
7:		∅
8:	19	∅
9:		∅
→ 10:		∅

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:	15	10
5:		∅
6:	28	∅
7:		∅
8:	19	∅
→ 9:		∅
10:	26	∅

Encadeamento Explícito sem Junção de Cadeias

Ao se inserir um registro com chave k :

1. se a posição $h(k)$ estiver livre, insere-se o registro nessa posição
2. se a posição $h(k)$ estiver ocupada por um registro com chave com mesmo endereço original de k , insere-se o novo registro na cadeia que se inicia nesta posição
3. senão, realoca-se o registro que estava na posição $h(k)$ para uma posição vazia e insere-se o novo registro em $h(k)$

Exemplo: 15, 28, 19, 26, 4, 30, 41, 18 - $h(k) = k \bmod m$, $m = 11$

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:	15	∅
5:		∅
6:	28	∅
7:		∅
8:	19	∅
9:		∅
→ 10:		∅

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:	15	10
5:		∅
6:	28	∅
7:		∅
8:	19	∅
→ 9:		∅
10:	26	∅

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:	15	10
5:		∅
6:	28	∅
→ 7:		∅
8:	19	∅
9:	4	∅
10:	26	9

Encadeamento Explícito sem Junção de Cadeias

Ao se inserir um registro com chave k :

1. se a posição $h(k)$ estiver livre, insere-se o registro nessa posição
2. se a posição $h(k)$ estiver ocupada por um registro com chave com mesmo endereço original de k , insere-se o novo registro na cadeia que se inicia nesta posição
3. senão, realoca-se o registro que estava na posição $h(k)$ para uma posição vazia e insere-se o novo registro em $h(k)$

Exemplo: 15, 28, 19, 26, 4, 30, 41, 18 - $h(k) = k \bmod m$, $m = 11$

End	Reg.	Pt
0:		\emptyset
1:		\emptyset
2:		\emptyset
3:		\emptyset
4:	15	10
→ 5:		\emptyset
6:	28	\emptyset
7:	30	\emptyset
8:	19	7
9:	4	\emptyset
10:	26	9

Encadeamento Explícito sem Junção de Cadeias

Ao se inserir um registro com chave k :

1. se a posição $h(k)$ estiver livre, insere-se o registro nessa posição
2. se a posição $h(k)$ estiver ocupada por um registro com chave com mesmo endereço original de k , insere-se o novo registro na cadeia que se inicia nesta posição
3. senão, realoca-se o registro que estava na posição $h(k)$ para uma posição vazia e insere-se o novo registro em $h(k)$

Exemplo: 15, 28, 19, 26, 4, 30, 41, 18 - $h(k) = k \bmod m$, $m = 11$

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:	15	10
→ 5:		∅
6:	28	∅
7:	30	∅
8:	19	7
9:	4	∅
10:	26	9

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
→ 3:		∅
4:	15	10
5:	41	∅
6:	28	∅
7:	30	5
8:	19	7
9:	4	∅
10:	26	9

Encadeamento Explícito sem Junção de Cadeias

Ao se inserir um registro com chave k :

1. se a posição $h(k)$ estiver livre, insere-se o registro nessa posição
2. se a posição $h(k)$ estiver ocupada por um registro com chave com mesmo endereço original de k , insere-se o novo registro na cadeia que se inicia nesta posição
3. senão, realoca-se o registro que estava na posição $h(k)$ para uma posição vazia e insere-se o novo registro em $h(k)$

Exemplo: 15, 28, 19, 26, 4, 30, 41, 18 - $h(k) = k \bmod m$, $m = 11$

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:	15	10
→ 5:		∅
6:	28	∅
7:	30	∅
8:	19	7
9:	4	∅
10:	26	9

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
→ 3:		∅
4:	15	10
5:	41	∅
6:	28	∅
7:	30	5
8:	19	7
9:	4	∅
10:	26	9

End	Reg.	Pt
0:		∅
1:		∅
→ 2:		∅
3:	30	5
4:	15	10
5:	41	∅
6:	28	∅
7:	18	∅
8:	19	3
9:	4	∅
10:	26	9

Encadeamento Explícito sem Junção de Cadeias

Média de acessos:

$$\frac{1+1+1+2+3+2+3+1}{8} = 1,75$$

Como não há junção de cadeias, este método reflete o método com alocação dinâmica e, portanto, possui os mesmos resultados de eficiência.

Tempo esperado para consulta: $O(1 + \alpha)$, $\alpha = \frac{n}{m}$.

Como $n \leq m$, o tempo esperado é $O(1)$.

Exercício

Qual seria o estado final de um arquivo usando hash com endereçamento explícito sem junção de cadeias com $m = 11$ e $h(k) = k \bmod m$, após a inserção da seguinte sequência de chaves:

16, 19, 38, 10, 31, 11, 27, 13

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:		∅
5:		∅
6:		∅
7:		∅
8:		∅
9:		∅
→ 10:		∅

Exercício

Qual seria o estado final de um arquivo usando hash com endereçamento explícito sem junção de cadeias com $m = 11$ e $h(k) = k \bmod m$, após a inserção da seguinte sequência de chaves:

16, 19, 38, 10, 31, 11, 27, 13

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:		∅
5:		∅
6:		∅
7:		∅
8:		∅
9:		∅
→ 10:		∅

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:		∅
5:	16	∅
6:		∅
7:		∅
8:		∅
9:		∅
→ 10:		∅

Exercício

Qual seria o estado final de um arquivo usando hash com endereçamento explícito sem junção de cadeias com $m = 11$ e $h(k) = k \bmod m$, após a inserção da seguinte sequência de chaves:

16, 19, 38, 10, 31, 11, 27, 13

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:		∅
5:		∅
6:		∅
7:		∅
8:		∅
9:		∅
→ 10:		∅

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:		∅
5:	16	∅
6:		∅
7:		∅
8:		∅
9:		∅
→ 10:		∅

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:		∅
5:	16	∅
6:		∅
7:		∅
8:	19	∅
9:		∅
→ 10:		∅

Exercício

Qual seria o estado final de um arquivo usando hash com endereçamento explícito sem junção de cadeias com $m = 11$ e $h(k) = k \bmod m$, após a inserção da seguinte sequência de chaves:

16, 19, 38, 10, 31, 11, 27, 13

End	Reg.	Pt
0:		\emptyset
1:		\emptyset
2:		\emptyset
3:		\emptyset
4:		\emptyset
5:	16	10
6:		\emptyset
7:		\emptyset
8:	19	\emptyset
→ 9:		\emptyset
10:	38	\emptyset

Exercício

Qual seria o estado final de um arquivo usando hash com endereçamento explícito sem junção de cadeias com $m = 11$ e $h(k) = k \bmod m$, após a inserção da seguinte sequência de chaves:

16, 19, 38, 10, 31, 11, 27, 13

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:		∅
5:	16	10
6:		∅
7:		∅
8:	19	∅
→ 9:		∅
10:	38	∅

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:		∅
5:	16	9
6:		∅
→ 7:		∅
8:	19	∅
9:	38	∅
10:	10	∅

Exercício

Qual seria o estado final de um arquivo usando hash com endereçamento explícito sem junção de cadeias com $m = 11$ e $h(k) = k \bmod m$, após a inserção da seguinte sequência de chaves:

16, 19, 38, 10, 31, 11, 27, 13

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:		∅
5:	16	10
6:		∅
7:		∅
8:	19	∅
→ 9:		∅
10:	38	∅

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:		∅
5:	16	9
6:		∅
→ 7:		∅
8:	19	∅
9:	38	∅
10:	10	∅

End	Reg.	Pt
0:		∅
1:		∅
2:		∅
3:		∅
4:		∅
5:	16	7
→ 6:		∅
7:	38	∅
8:	19	∅
9:	31	∅
10:	10	∅

Exercício

Qual seria o estado final de um arquivo usando hash com endereçamento explícito sem junção de cadeias com $m = 11$ e $h(k) = k \bmod m$, após a inserção da seguinte sequência de chaves:

16, 19, 38, 10, 31, 11, 27, 13

End	Reg.	Pt
0:	11	\emptyset
1:		\emptyset
2:		\emptyset
3:		\emptyset
4:		\emptyset
5:	16	7
→ 6:		\emptyset
7:	38	\emptyset
8:	19	\emptyset
9:	31	\emptyset
10:	10	\emptyset

Exercício

Qual seria o estado final de um arquivo usando hash com endereçamento explícito sem junção de cadeias com $m = 11$ e $h(k) = k \bmod m$, após a inserção da seguinte sequência de chaves:

16, 19, 38, 10, 31, 11, 27, 13

End	Reg.	Pt
0:	11	∅
1:		∅
2:		∅
3:		∅
4:		∅
5:	16	7
→ 6:		∅
7:	38	∅
8:	19	∅
9:	31	∅
10:	10	∅

End	Reg.	Pt
0:	11	∅
1:		∅
2:		∅
3:		∅
→ 4:		∅
5:	16	7
6:	27	∅
7:	38	6
8:	19	∅
9:	31	∅
10:	10	∅

Exercício

Qual seria o estado final de um arquivo usando hash com endereçamento explícito sem junção de cadeias com $m = 11$ e $h(k) = k \bmod m$, após a inserção da seguinte sequência de chaves:

16, 19, 38, 10, 31, 11, 27, 13

End	Reg.	Pt
0:	11	∅
1:		∅
2:		∅
3:		∅
4:		∅
5:	16	7
→ 6:		∅
7:	38	∅
8:	19	∅
9:	31	∅
10:	10	∅

End	Reg.	Pt
0:	11	∅
1:		∅
2:		∅
3:		∅
→ 4:		∅
5:	16	7
6:	27	∅
7:	38	6
8:	19	∅
9:	31	∅
10:	10	∅

End	Reg.	Pt
0:	11	∅
1:		∅
2:	13	∅
3:		∅
→ 4:		∅
5:	16	7
6:	27	∅
7:	38	6
8:	19	∅
9:	31	∅
10:	10	∅

Exercício

Qual seria o estado final de um arquivo usando hash com endereçamento explícito sem junção de cadeias com $m = 11$ e $h(k) = k \bmod m$, após a inserção da seguinte sequência de chaves:

16, 19, 38, 10, 31, 11, 27, 13

End	Reg.	Pt
0:	11	∅
1:		∅
2:		∅
3:		∅
4:		∅
5:	16	7
→ 6:		∅
7:	38	∅
8:	19	∅
9:	31	∅
10:	10	∅

End	Reg.	Pt
0:	11	∅
1:		∅
2:		∅
3:		∅
→ 4:		∅
5:	16	7
6:	27	∅
7:	38	6
8:	19	∅
9:	31	∅
10:	10	∅

End	Reg.	Pt
0:	11	∅
1:		∅
2:	13	∅
3:		∅
→ 4:		∅
5:	16	7
6:	27	∅
7:	38	6
8:	19	∅
9:	31	∅
10:	10	∅

Média de acessos: $\frac{1+1+2+1+1+1+3+1}{8} = 1,38$