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

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
- ightharpoonup Usa-se h(k) para se definir onde k ficará armazenada no arquivo

	⊨na	Cnave	into	
	0:	k_1	r_1	
	1:	-	-	
$h(k) \to \{0, 1,, m-1\}$	2:	k_2	<i>r</i> ₂	
	m-1:	k _n	r _n	

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

$$h(k)
ightarrow \{0,1,...,m-1\}$$
 End Chave Info
$$0: \begin{array}{c|cccc} k_1 & r_1 \\ 1: & - & - \\ 2: & k_2 & r_2 \\ ... & ... & ... \\ m-1: & k_n & r_n \end{array}$$

▶ 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

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

$$h(k)
ightarrow \{0,1,...,m-1\}$$
 End Chave Info
$$0: \begin{array}{c|cccc} k_1 & r_1 \\ 1: & - & - \\ 2: & k_2 & r_2 \\ ... & ... & ... \\ m-1: & k_n & r_n \end{array}$$

- 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

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	<i>r</i> ₂
m - 1:	k _n	r _n

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	<i>r</i> ₂
	• • •	
m - 1:	k _n	r _n

Qual o problema desta abordagem?

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	<i>r</i> ₂
2:	<i>k</i> ₃	<i>r</i> ₃
• • • •	• • •	
n - 1:	k _n	r _n

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	<i>r</i> ₂
2:	<i>k</i> ₃	<i>r</i> 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.

Hashing

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:

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

Hashing

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:

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

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



- 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 $h(k) = k \mod m$.

End	List	as de	registros
0:	Ø		
1:	Ø		
2:	Ø		
3:	Ø		
4:	Ø		
5:	Ø		
6:	Ø		
7:	Ø		
8:	Ø		
Q٠	()		

10:

- 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 $h(k) = k \mod m$.

End	Listas de registros	End	Lista	s de re	gistros
0:	Ø	0:	Ø		
1:	Ø	1:	Ø		
1: 2: 3: 4: 5: 6:	Ø	2:	Ø		
3:	Ø	3:	Ø		
4:	Ø	4:	Ø		
5:	Ø	5:	27	$\rightarrow \emptyset$	
6:	Ø	6:	Ø		
7:	Ø	7:	Ø		
8:	Ø	8:	Ø		
9:	Ø	9:	Ø		
10:	Ø	10:	Ø		

- 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 $h(k) = k \mod m$.

End	Listas de registros	End	Lista	s de registros	End	Lista	s de registros
0:	Ø	0:	0	o de regiones	0:	Ø	io de regiones
1:	Ø	1:	Ø		1:	Ø	
2:	Ø	2:	Ø		2:	Ø	
3:	Ø	3:	Ø		3:	Ø	
4:	Ø	4:	Ø		4:	Ø	
5:	Ø	5:	27	$\rightarrow \emptyset$	5:	27	$\rightarrow \emptyset$
6:	Ø	6:	Ø		6:	Ø	
7:	Ø	7:	Ø		7:	18	$\rightarrow \emptyset$
8:	Ø	8:	Ø		8:	Ø	
9:	Ø	9:	Ø		9:	Ø	
10:	Ø	10:	Ø		10:	Ø	

- 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 $h(k) = k \mod m$.

End	Lista	is de	registros
0:	Ø		
1:	Ø		
2:	Ø		
3:	Ø		
4:	Ø		
5:	27	\rightarrow	Ø
6:	Ø		
7:	18	\rightarrow	$29 \rightarrow \emptyset$
8:	Ø		
9:	Ø		
10:	Ø		

- 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 $h(k) = k \mod m$.

End	Lista	s de registros	End	Lista	as de registros
0:	0	ls de registros	0:	0	
1:	Ø		1:	0	
2:	0		2:	(A)	
3:	W O			W O	
	W		3:	W	
4:	Ø	_	4:	W	_
5:	27	$\rightarrow \emptyset$	5:	27	$\rightarrow \emptyset$
6:	Ø		6:	Ø	
7:	18	$ ightarrow$ 29 $ ightarrow$ \emptyset	7:	18	\rightarrow 29 \rightarrow \emptyset
8:	Ø		8:	Ø	
9:	Ø		9:	Ø	
10:	Ø		10:	43	$\rightarrow \emptyset$

- 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 $h(k) = k \mod m$.

End	Lista	s de registros	End	Lista	s de registros
0:	Ø		0:	Ø	
1:	Ø		1:	Ø	
2:	Ø		2:	Ø	
3:	Ø		3:	Ø	
4:	Ø		4:	Ø	
5:	27	$\rightarrow \emptyset$	5:	27	$\rightarrow \emptyset$
6:	Ø		6:	Ø	
7:	18	$ ightarrow$ 29 $ ightarrow$ \emptyset	7:	18	$ ightarrow$ 29 $ ightarrow$ \emptyset
8:	Ø		8:	Ø	
9:	Ø		9:	Ø	
10:	Ø		10:	43	$\rightarrow \emptyset$

End	Lista	s de registros
0:	77	$\rightarrow \emptyset$
1:	Ø	
2:	Ø	
3:	Ø	
4:	Ø	
5:	27	$\rightarrow \emptyset$
6:	Ø	
7:	18	$ ightarrow$ 29 $ ightarrow$ \emptyset
8:	Ø	
9:	Ø	
10:	43	$\rightarrow \emptyset$

- ▶ 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 $h(k) = k \mod m$.

End	Listas de re	egistros
0:	$77 \rightarrow \emptyset$	
1:	Ø	
2:	$13 \rightarrow \emptyset$	
3:	Ø	
4:	Ø	
5:	27 → 1	$6 o \emptyset$
6:	Ø	
7:	18 → 2°	$9 \rightarrow \emptyset$
8:	Ø	
9:	Ø	
10:	43 → ∅	

- ▶ 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 $h(k) = k \mod m$.

	1111		12.4
End	Listas de registros	End	Listas de registros
0:	$77 \rightarrow \emptyset$	0:	77 $\rightarrow \emptyset$
1:	Ø	1:	Ø
2:	13 → Ø	2:	13 → Ø
3:	Ø	3:	Ø
4:	Ø	4:	Ø
5:	$27 \rightarrow 16 \rightarrow \emptyset$	5:	$27 \rightarrow 16 \rightarrow \emptyset$
6:	Ø	6:	Ø
7:	18 → 29 → Ø	7:	$\boxed{18 \rightarrow 29 \rightarrow 40 \rightarrow \emptyset}$
8:	Ø	8:	Ø
9:	Ø	9:	Ø
10:	43 → ∅	10:	43 → ∅
		_	

- 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 $h(k) = k \mod m$.

End	Lista	s de registros	End	Listas de registros
0:	77	$\rightarrow \emptyset$	0:	$77 \rightarrow \emptyset$
1:	Ø		1:	Ø
2:	13	$\rightarrow \emptyset$	2:	$13 \rightarrow \emptyset$
3:	Ø		3:	Ø
4:	Ø		4:	Ø
5:	27	$ ightarrow$ 16 $ ightarrow$ \emptyset	5:	$27 \rightarrow 16 \rightarrow \emptyset$
6:	Ø		6:	Ø
7:	18	$ ightarrow$ 29 $ ightarrow$ \emptyset	7:	$\begin{array}{ c c c c c }\hline 18 & \rightarrow 29 \rightarrow 40 \rightarrow \emptyset \\ \hline \end{array}$
8:	Ø		8:	Ø
9:	Ø		9:	Ø
10:	43	$\rightarrow \emptyset$	10:	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?

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

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

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

End	Reg.	Pt	End	Reg.	Pt
0:		Ø	0:		Ø
1:		Ø	1:		Ø
2:		Ø	2:		Ø
3:		Ø	2: 3: 4:		Ø
4:		Ø	4:		Ø
1: 2: 3: 4: 5: 6: 7: 8: 9:		Ø	5:	27	Ø
6:		Ø	6:		Ø
7:		Ø	7:		Ø
8:		Ø	8:		Ø
9:		Ø	9:		Ø
\rightarrow 10:		Ø	ightarrow 10:		Ø

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Vantagens e desvantagens?

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

- ightharpoonup 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

- ightharpoonup 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?

Ao se inserir um registro com chave k:

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

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

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)

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)

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

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)

End	Reg.	Pt	End	Reg.	Pt
0:		Ø	0:		Ø
1:		Ø	1:		Ø
1: 2: 3: 4: 5: 6:		Ø	1: 2: 3: 4: 5: 6:		Ø
3:		Ø	3:		Ø
4:		Ø	4:	15	Ø
5:		Ø	5:		Ø
		Ø			Ø
7: 8: 9:		Ø	7: 8: 9:		Ø
8:		Ø	8:		Ø
9:		Ø	9:		Ø
\rightarrow 10:		Ø	\rightarrow 10:		Ø

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)

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

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

End	Reg.	Pt
0:		Ø
1:		Ø
2:		Ø
3:		Ø
4:	15	Ø
5:		Ø
6:	28	Ø
7:		Ø
8:		Ø
9:		Ø
→ 10:	↓ □→	Ø,

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)

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

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)

End	Reg.	Pt	End	Reg.	Pt
	iveg.	1 1		rteg.	1 L
0:		Ø	0:		Ø
1:		Ø	1:		Ø
2:		Ø	1: 2: 3:		Ø
2: 3:		Ø	3:		Ø
4:	15	Ø	4: 5:	15	10
5:		Ø	5:		Ø
6:	28	Ø	6:	28	Ø
7:		Ø	7:		Ø
8:	19	Ø	8:	19	Ø
9:		Ø	\rightarrow 9:		Ø
→ 10:		Ø	10:	26	Ø

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)

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

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)

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

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)

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

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

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)

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

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

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

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 $n \le m$, o tempo esperado é O(1).

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

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

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

End	Reg.	Pt	End	Reg.	Pt
0:		Ø	0:		Ø
1:		Ø	1:		Ø
2:		Ø	2: 3: 4:		Ø
3:		Ø	3:		Ø
4:		Ø	4:		Ø
5:		Ø	5: 6:	16	Ø
6:		Ø			Ø
7:		Ø	7: 8:		Ø
8:		Ø	8:		Ø
9:		Ø	9:		Ø
\rightarrow 10:		Ø	\rightarrow 10:		Ø

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

End	Reg.	Ρt
0:	11	Ø
1:		Ø
2:		Ø
3:		Ø
4:		Ø
5:	16	7
\rightarrow 6:		Ø
7:	38	Ø
8:	19	Ø
9:	31	Ø
10:	10	Ø

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

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

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

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

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