

Tutorium

Grundlagen: Algorithmen und Datenstrukturen

Übungsblatt Woche 6

Aufgabe 6.1 – Universelles Hashing

Die Namen $\{\text{Adalfuns, Berahthraban, Chlothar, Dagoberta, Egino}\}$ sollen in einer Hash-Tabelle der Größe $m = 4$ untergebracht werden. Es seien folgende Hashfunktionen gegeben:

f_1 :	Adalfuns $\mapsto 4$	Berahthraban $\mapsto 2$	Chlothar $\mapsto 2$	Dagoberta $\mapsto 1$	Egino $\mapsto 4$
f_2 :	Adalfuns $\mapsto 3$	Berahthraban $\mapsto 4$	Chlothar $\mapsto 2$	Dagoberta $\mapsto 3$	Egino $\mapsto 4$
f_3 :	Adalfuns $\mapsto 2$	Berahthraban $\mapsto 2$	Chlothar $\mapsto 4$	Dagoberta $\mapsto 1$	Egino $\mapsto 1$
f_4 :	Adalfuns $\mapsto 1$	Berahthraban $\mapsto 3$	Chlothar $\mapsto 3$	Dagoberta $\mapsto 4$	Egino $\mapsto 4$
<hr/>					
g_1 :	Adalfuns $\mapsto 1$	Berahthraban $\mapsto 1$	Chlothar $\mapsto 3$	Dagoberta $\mapsto 2$	Egino $\mapsto 3$
g_2 :	Adalfuns $\mapsto 2$	Berahthraban $\mapsto 4$	Chlothar $\mapsto 2$	Dagoberta $\mapsto 3$	Egino $\mapsto 4$
g_3 :	Adalfuns $\mapsto 4$	Berahthraban $\mapsto 4$	Chlothar $\mapsto 1$	Dagoberta $\mapsto 4$	Egino $\mapsto 2$
g_4 :	Adalfuns $\mapsto 3$	Berahthraban $\mapsto 1$	Chlothar $\mapsto 2$	Dagoberta $\mapsto 3$	Egino $\mapsto 3$
g_5 :	Adalfuns $\mapsto 4$	Berahthraban $\mapsto 2$	Chlothar $\mapsto 2$	Dagoberta $\mapsto 2$	Egino $\mapsto 3$

Aufgabe 6.1 – Universelles Hashing

f_1 :	Adalfuns \mapsto 4	Berahthraban \mapsto 2	Chlothar \mapsto 2	Dagoberta \mapsto 1	Egino \mapsto 4
f_2 :	Adalfuns \mapsto 3	Berahthraban \mapsto 4	Chlothar \mapsto 2	Dagoberta \mapsto 3	Egino \mapsto 4
f_3 :	Adalfuns \mapsto 2	Berahthraban \mapsto 2	Chlothar \mapsto 4	Dagoberta \mapsto 1	Egino \mapsto 1
f_4 :	Adalfuns \mapsto 1	Berahthraban \mapsto 3	Chlothar \mapsto 3	Dagoberta \mapsto 4	Egino \mapsto 4
g_1 :	Adalfuns \mapsto 1	Berahthraban \mapsto 1	Chlothar \mapsto 3	Dagoberta \mapsto 2	Egino \mapsto 3
g_2 :	Adalfuns \mapsto 2	Berahthraban \mapsto 4	Chlothar \mapsto 2	Dagoberta \mapsto 3	Egino \mapsto 4
g_3 :	Adalfuns \mapsto 4	Berahthraban \mapsto 4	Chlothar \mapsto 1	Dagoberta \mapsto 4	Egino \mapsto 2
g_4 :	Adalfuns \mapsto 3	Berahthraban \mapsto 1	Chlothar \mapsto 2	Dagoberta \mapsto 3	Egino \mapsto 3
g_5 :	Adalfuns \mapsto 4	Berahthraban \mapsto 2	Chlothar \mapsto 2	Dagoberta \mapsto 2	Egino \mapsto 3

Paar	f_1	f_2	f_3	f_4	g_1	g_2	g_3	g_4	g_5
Adalfuns/Berahthraban									
Adalfuns/Chlothar									
Adalfuns/Dagoberta									
Adalfuns/Egino									
Berahthraban/Chlothar									
Berahthraban/Dagoberta									
Berahthraban/Egino									
Chlothar/Dagoberta									
Chlothar/Egino									
Dagoberta/Egino									

Aufgabe 6.1 – Universelles Hashing

f_1 :	Adalfuns $\mapsto 4$	Berahthraban $\mapsto 2$	Chlothar $\mapsto 2$	Dagoberta $\mapsto 1$	Egino $\mapsto 4$
f_2 :	Adalfuns $\mapsto 3$	Berahthraban $\mapsto 4$	Chlothar $\mapsto 2$	Dagoberta $\mapsto 3$	Egino $\mapsto 4$
f_3 :	Adalfuns $\mapsto 2$	Berahthraban $\mapsto 2$	Chlothar $\mapsto 4$	Dagoberta $\mapsto 1$	Egino $\mapsto 1$
f_4 :	Adalfuns $\mapsto 1$	Berahthraban $\mapsto 3$	Chlothar $\mapsto 3$	Dagoberta $\mapsto 4$	Egino $\mapsto 4$
g_1 :	Adalfuns $\mapsto 1$	Berahthraban $\mapsto 1$	Chlothar $\mapsto 3$	Dagoberta $\mapsto 2$	Egino $\mapsto 3$
g_2 :	Adalfuns $\mapsto 2$	Berahthraban $\mapsto 4$	Chlothar $\mapsto 2$	Dagoberta $\mapsto 3$	Egino $\mapsto 4$
g_3 :	Adalfuns $\mapsto 4$	Berahthraban $\mapsto 4$	Chlothar $\mapsto 1$	Dagoberta $\mapsto 4$	Egino $\mapsto 2$
g_4 :	Adalfuns $\mapsto 3$	Berahthraban $\mapsto 1$	Chlothar $\mapsto 2$	Dagoberta $\mapsto 3$	Egino $\mapsto 3$
g_5 :	Adalfuns $\mapsto 4$	Berahthraban $\mapsto 2$	Chlothar $\mapsto 2$	Dagoberta $\mapsto 2$	Egino $\mapsto 3$

Paar	f_1	f_2	f_3	f_4	g_1	g_2	g_3	g_4	g_5
Adalfuns/Berahthraban			x		x		x		
Adalfuns/Chlothar						x			
Adalfuns/Dagoberta		x					x	x	
Adalfuns/Egino	x							x	
Berahthraban/Chlothar	x			x					x
Berahthraban/Dagoberta							x		x
Berahthraban/Egino		x				x			
Chlothar/Dagoberta									x
Chlothar/Egino					x				
Dagoberta/Egino			x	x				x	

Aufgabe 6.1 – Universelles Hashing

Paar	f_1	f_2	f_3	f_4	g_1	g_2	g_3	g_4	g_5
Adalfuns/Berahthraban			x		x		x		
Adalfuns/Chlothar						x			
Adalfuns/Dagoberta		x					x	x	
Adalfuns/Egino	x							x	
Berahthraban/Chlothar	x			x					x
Berahthraban/Dagoberta							x		x
Berahthraban/Egino		x				x			
Chlothar/Dagoberta									x
Chlothar/Egino					x				
Dagoberta/Egino			x	x				x	

$$\text{c-Universell wenn: } \frac{|\{f \in \mathcal{H} : f(x) = f(y)\}|}{|\mathcal{H}|} \leq \frac{c}{m}.$$

\nearrow Anzahl Kollisionen in f
 \searrow Anzahl Hashfunktionen in \mathcal{H}

Aufgabe 6.1 – Universelles Hashing

Paar	f_1	f_2	f_3	f_4	g_1	g_2	g_3	g_4	g_5
Adalfuns/Berahthraban			x		x		x		
Adalfuns/Chlothar						x			
Adalfuns/Dagoberta		x					x	x	
Adalfuns/Egino	x							x	
Berahthraban/Chlothar	x			x					x
Berahthraban/Dagoberta							x		x
Berahthraban/Egino		x				x			
Chlothar/Dagoberta									x
Chlothar/Egino					x				
Dagoberta/Egino			x	x				x	

- (a) Geben Sie für die Familie $\mathcal{H}_1 = \{f_1, f_2, f_3, f_4\}$ das kleinste c an, so dass \mathcal{H}_1 c -universell ist.
- (b) Finden Sie eine möglichst kleine Familie $\mathcal{H}_2 \subseteq \{g_1, g_2, g_3, g_4, g_5\}$, die 1-universell ist. Untermauern Sie Ihre Aussagen mit glaubwürdigen Argumenten.

Aufgabe 6.2 – Die Perfekte Hashtabelle

Zur Erinnerung: Beim statischen perfekten Hashing wird in der ersten Stufe so lange eine Hashfunktion $h : \text{Key} \rightarrow \{0, \dots, \lceil \sqrt{2cn} \rceil - 1\}$ aus einer c -universellen Familie $H_{\lceil \sqrt{2cn} \rceil}$ gezogen, bis für die gezogene Hashfunktion die Zahl $C(h)$ der Kollisionen maximal $\sqrt{2n}$ beträgt. Letzteres ist eine hinreichende Voraussetzung dafür, dass die zweite Stufe in erwarteter linearer Laufzeit ablaufen kann. Die Laufzeit für die erste Stufe ist ebenfalls erwartet linear.

Alle Schlüssel, die durch h auf dieselbe Position abgebildet werden, gehören zum selben Bucket. Wenn b_ℓ Schlüssel zu Bucket B_ℓ mit $\ell \in \{0, \dots, \lceil \sqrt{2cn} \rceil - 1\}$ gehören, dann hat das Bucket die Größe $m_\ell = cb_\ell(b_\ell - 1) + 1$.

In der zweiten Stufe wird für jedes Bucket B_ℓ so lange eine Hashfunktion h_ℓ einer c -universellen Familie H_{m_ℓ} gezogen, bis h_ℓ die Schlüssel von Bucket B_ℓ injektiv abbildet.

Bei dieser Aufgabe sind die Hashfunktionen bereits gegeben und müssen nicht erst gezogen werden.

Aufgabe 6.2 – Die Perfekte Hashtabelle

Konstruieren Sie eine statische perfekte Hashtabelle für die Elemente:

$$\begin{array}{cccccc} (16, 10, 11) & (8, 2, 15) & (7, 12, 8) & (1, 10, 3) & (13, 11, 14) & (6, 11, 14) \\ (7, 3, 16) & (2, 2, 8) & (10, 5, 15) & (7, 3, 14) & (2, 10, 1) & (14, 11, 6) \end{array}$$

Jedes Element x besteht aus den Stellen (x_0, x_1, x_2) . Verwenden Sie jeweils passend eine der Hashfunktionen:

$$\begin{aligned} & (\sum_{i=0}^2 2^i x_i) \bmod 17 \\ & (\sum_{i=0}^2 a_i x_i) \bmod 7 \text{ mit } \mathbf{a} = (0, 0, 1) \text{ oder } \mathbf{a} = (6, 6, 2) \\ & (\sum_{i=0}^2 a_i x_i) \bmod 3 \text{ mit } \mathbf{a} = (1, 0, 0) \text{ oder } \mathbf{a} = (0, 2, 2). \end{aligned}$$

Aufgabe 6.2 – Die Perfekte Hashtabelle

12	4	12	16	6	16
(16, 10, 11)	(8, 2, 15)	(7, 12, 8)	(1, 10, 3)	(13, 11, 14)	(6, 11, 14)
(7, 3, 16)	(2, 2, 8)	(10, 5, 15)	(7, 3, 14)	(2, 10, 1)	(14, 11, 6)
9	4	12	1	9	9

$$(\sum_{i=0}^2 2^i x_i) \bmod 17$$

$$(\sum_{i=0}^2 a_i x_i) \bmod 7 \text{ mit } \mathbf{a} = (0, 0, 1) \text{ oder } \mathbf{a} = (6, 6, 2)$$

$$(\sum_{i=0}^2 a_i x_i) \bmod 3 \text{ mit } \mathbf{a} = (1, 0, 0) \text{ oder } \mathbf{a} = (0, 2, 2).$$

Stufe 1:

Stufe 2:

$\sqrt{2} \cdot n$ Kollisionen
 $\sqrt{2} \cdot c \cdot n$ Plätze in
 erster Stufe
 $n=12$

Aufgabe 6.2 – Die Perfekte Hashtabelle

$(16, 10, 11)$ $(8, 2, 15)$ $(7, 12, 8)$ $(1, 10, 3)$ $(13, 11, 14)$ $(6, 11, 14)$
 $(7, 3, 16)$ $(2, 2, 8)$ $(10, 5, 15)$ $(7, 3, 14)$ $(2, 10, 1)$ $(14, 11, 6)$

$$(\sum_{i=0}^2 2^i x_i) \bmod 17$$

$(\sum_{i=0}^2 a_i x_i) \bmod 7$ mit $\mathbf{a} = (0, 0, 1)$ oder $\mathbf{a} = (6, 6, 2)$

$(\sum_{i=0}^2 a_i x_i) \bmod 3$ mit $\mathbf{a} = (1, 0, 0)$ oder $\mathbf{a} = (0, 2, 2)$.

Element	Bucket	Element	Bucket	Position in Bucket
$(7, 3, 14)$	1	$(7, 3, 14)$	1	0
$(2, 2, 8)$	4	$(2, 2, 8)$	4	2
$(8, 2, 15)$	4	$(8, 2, 15)$	4	1
$(13, 11, 14)$	6	$(13, 11, 14)$	6	0
$(2, 10, 1)$	9	$(2, 10, 1)$	9	1
$(14, 11, 6)$	9	$(14, 11, 6)$	9	6
$(7, 3, 16)$	9	$(7, 3, 16)$	9	2
$(16, 10, 11)$	12	$(16, 10, 11)$	12	3
$(7, 12, 8)$	12	$(7, 12, 8)$	12	4
$(10, 5, 15)$	12	$(10, 5, 15)$	12	1
$(1, 10, 3)$	16	$(1, 10, 3)$	16	1
$(6, 11, 14)$	16	$(6, 11, 14)$	16	0

Aufgabe 6.3 – Linear Probing

Die Größe der Hashtabelle ist dabei jeweils $m = 13$. Führen Sie die folgenden Operationen aus:

```
insert 16, 3, 12, 17, 29, 10, 24
delete 16
insert 5, 1, 15
delete 10
insert 14
delete 1
```

Verwenden Sie die Hashfunktion

$$h(x) = 3x \bmod 13.$$

Bei dieser Aufgabe sind die Schlüssel der Elemente die Elemente selbst.

Beim Löschen soll die dritte Methode aus der Vorlesung verwendet werden, d.h. die Wiederherstellung der folgenden Invariante: Für jedes Element e in der Hashtabelle mit Schlüssel $k(e)$, aktueller Position j und optimaler Position $i = h(k(e))$ sind alle Positionen $i, (i + 1) \bmod m, (i + 2) \bmod m, \dots, j$ der Hashtabelle belegt. Bei dieser Aufgabe soll keine dynamische Größenanpassung der Hashtabelle stattfinden.

Invariante in einfach:
Zwischen der optimalen Position und der aktuellen Position darf kein Feld mehr frei sein

Aufgabe 6.3 – Linear Probing

1. Operation: `insert(16)` mit opt. Position: 9

Aufgabe 6.3 – Linear Probing

1. Operation: `insert(16)` mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
									16			

2. Operation: `insert(3)` mit opt. Position: 9

Aufgabe 6.3 – Linear Probing

1. Operation: **insert**(16) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
									16			

2. Operation: **insert**(3) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3		

3. Operation: **insert**(12) mit opt. Position: 10

Aufgabe 6.3 – Linear Probing

1. Operation: **insert**(16) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
									16			

2. Operation: **insert**(3) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3		

3. Operation: **insert**(12) mit opt. Position: 10

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3	12	

4. Operation: **insert**(17) mit opt. Position: 12

Aufgabe 6.3 – Linear Probing

1. Operation: **insert**(16) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
									16			

2. Operation: **insert**(3) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3		

3. Operation: **insert**(12) mit opt. Position: 10

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3	12	

4. Operation: **insert**(17) mit opt. Position: 12

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3	12	17

5. Operation: **insert**(29) mit opt. Position: 9

Aufgabe 6.3 – Linear Probing

1. Operation: **insert**(16) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
									16			

2. Operation: **insert**(3) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3		

3. Operation: **insert**(12) mit opt. Position: 10

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3	12	

4. Operation: **insert**(17) mit opt. Position: 12

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3	12	17

5. Operation: **insert**(29) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
29									16	3	12	17

6. Operation: **insert**(10) mit opt. Position: 4

Aufgabe 6.3 – Linear Probing

1. Operation: **insert**(16) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
									16			

2. Operation: **insert**(3) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3		

3. Operation: **insert**(12) mit opt. Position: 10

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3	12	

4. Operation: **insert**(17) mit opt. Position: 12

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3	12	17

5. Operation: **insert**(29) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
29									16	3	12	17

6. Operation: **insert**(10) mit opt. Position: 4

0	1	2	3	4	5	6	7	8	9	10	11	12
29				10					16	3	12	17

7. Operation: **insert**(24) mit opt. Position: 7

Aufgabe 6.3 – Linear Probing

1. Operation: **insert**(16) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
									16			

2. Operation: **insert**(3) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3		

3. Operation: **insert**(12) mit opt. Position: 10

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3	12	

4. Operation: **insert**(17) mit opt. Position: 12

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3	12	17

5. Operation: **insert**(29) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
29									16	3	12	17

6. Operation: **insert**(10) mit opt. Position: 4

0	1	2	3	4	5	6	7	8	9	10	11	12
29				10					16	3	12	17

7. Operation: **insert**(24) mit opt. Position: 7

0	1	2	3	4	5	6	7	8	9	10	11	12
29				10			24		16	3	12	17

8. Operation: **delete**(16) mit opt. Position: 9

Aufgabe 6.3 – Linear Probing

1. Operation: **insert**(16) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
									16			

2. Operation: **insert**(3) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3		

3. Operation: **insert**(12) mit opt. Position: 10

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3	12	

4. Operation: **insert**(17) mit opt. Position: 12

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3	12	17

5. Operation: **insert**(29) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
29									16	3	12	17

6. Operation: **insert**(10) mit opt. Position: 4

0	1	2	3	4	5	6	7	8	9	10	11	12
29				10					16	3	12	17

7. Operation: **insert**(24) mit opt. Position: 7

0	1	2	3	4	5	6	7	8	9	10	11	12
29				10			24		16	3	12	17

8. Operation: **delete**(16) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
				10			24		3	12	29	17

9. Operation: **insert**(5) mit opt. Position: 2

Aufgabe 6.3 – Linear Probing

1. Operation: **insert**(16) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
									16			

2. Operation: **insert**(3) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3		

3. Operation: **insert**(12) mit opt. Position: 10

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3	12	

4. Operation: **insert**(17) mit opt. Position: 12

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3	12	17

5. Operation: **insert**(29) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
29									16	3	12	17

6. Operation: **insert**(10) mit opt. Position: 4

0	1	2	3	4	5	6	7	8	9	10	11	12
29				10					16	3	12	17

7. Operation: **insert**(24) mit opt. Position: 7

0	1	2	3	4	5	6	7	8	9	10	11	12
29				10			24		16	3	12	17

8. Operation: **delete**(16) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
				10			24		3	12	29	17

9. Operation: **insert**(5) mit opt. Position: 2

0	1	2	3	4	5	6	7	8	9	10	11	12
		5		10			24		3	12	29	17

10. Operation: **insert**(1) mit opt. Position: 3

Aufgabe 6.3 – Linear Probing

1. Operation: **insert**(16) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
									16			

2. Operation: **insert**(3) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3		

3. Operation: **insert**(12) mit opt. Position: 10

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3	12	

4. Operation: **insert**(17) mit opt. Position: 12

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3	12	17

5. Operation: **insert**(29) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
29									16	3	12	17

6. Operation: **insert**(10) mit opt. Position: 4

0	1	2	3	4	5	6	7	8	9	10	11	12
29				10					16	3	12	17

7. Operation: **insert**(24) mit opt. Position: 7

0	1	2	3	4	5	6	7	8	9	10	11	12
29				10			24		16	3	12	17

8. Operation: **delete**(16) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
				10			24		3	12	29	17

9. Operation: **insert**(5) mit opt. Position: 2

0	1	2	3	4	5	6	7	8	9	10	11	12
		5		10			24		3	12	29	17

10. Operation: **insert**(1) mit opt. Position: 3

0	1	2	3	4	5	6	7	8	9	10	11	12
		5	1	10			24		3	12	29	17

11. Operation: **insert**(15) mit opt. Position: 6

Aufgabe 6.3 – Linear Probing

1. Operation: **insert**(16) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
									16			

2. Operation: **insert**(3) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3		

3. Operation: **insert**(12) mit opt. Position: 10

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3	12	

4. Operation: **insert**(17) mit opt. Position: 12

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3	12	17

5. Operation: **insert**(29) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
29									16	3	12	17

6. Operation: **insert**(10) mit opt. Position: 4

0	1	2	3	4	5	6	7	8	9	10	11	12
29				10					16	3	12	17

7. Operation: **insert**(24) mit opt. Position: 7

0	1	2	3	4	5	6	7	8	9	10	11	12
29				10			24		16	3	12	17

8. Operation: **delete**(16) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
				10			24		3	12	29	17

9. Operation: **insert**(5) mit opt. Position: 2

0	1	2	3	4	5	6	7	8	9	10	11	12
		5		10			24		3	12	29	17

10. Operation: **insert**(1) mit opt. Position: 3

0	1	2	3	4	5	6	7	8	9	10	11	12
		5	1	10			24		3	12	29	17

11. Operation: **insert**(15) mit opt. Position: 6

0	1	2	3	4	5	6	7	8	9	10	11	12
		5	1	10		15	24		3	12	29	17

12. Operation: **delete**(10) mit opt. Position: 4

Aufgabe 6.3 – Linear Probing

1. Operation: **insert**(16) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
									16			

2. Operation: **insert**(3) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3		

3. Operation: **insert**(12) mit opt. Position: 10

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3	12	

4. Operation: **insert**(17) mit opt. Position: 12

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3	12	17

5. Operation: **insert**(29) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
29									16	3	12	17

6. Operation: **insert**(10) mit opt. Position: 4

0	1	2	3	4	5	6	7	8	9	10	11	12
29				10					16	3	12	17

7. Operation: **insert**(24) mit opt. Position: 7

0	1	2	3	4	5	6	7	8	9	10	11	12
29				10			24		16	3	12	17

8. Operation: **delete**(16) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
				10			24		3	12	29	17

9. Operation: **insert**(5) mit opt. Position: 2

0	1	2	3	4	5	6	7	8	9	10	11	12
		5		10			24		3	12	29	17

10. Operation: **insert**(1) mit opt. Position: 3

0	1	2	3	4	5	6	7	8	9	10	11	12
		5	1	10			24		3	12	29	17

11. Operation: **insert**(15) mit opt. Position: 6

0	1	2	3	4	5	6	7	8	9	10	11	12
		5	1	10		15	24		3	12	29	17

12. Operation: **delete**(10) mit opt. Position: 4

0	1	2	3	4	5	6	7	8	9	10	11	12
		5	1			15	24		3	12	29	17

13. Operation: **insert**(14) mit opt. Position: 3

Aufgabe 6.3 – Linear Probing

1. Operation: **insert**(16) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
									16			

2. Operation: **insert**(3) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3		

3. Operation: **insert**(12) mit opt. Position: 10

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3	12	

4. Operation: **insert**(17) mit opt. Position: 12

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3	12	17

5. Operation: **insert**(29) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
29									16	3	12	17

6. Operation: **insert**(10) mit opt. Position: 4

0	1	2	3	4	5	6	7	8	9	10	11	12
29				10					16	3	12	17

7. Operation: **insert**(24) mit opt. Position: 7

0	1	2	3	4	5	6	7	8	9	10	11	12
29				10			24		16	3	12	17

8. Operation: **delete**(16) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
				10			24		3	12	29	17

9. Operation: **insert**(5) mit opt. Position: 2

0	1	2	3	4	5	6	7	8	9	10	11	12
		5		10			24		3	12	29	17

10. Operation: **insert**(1) mit opt. Position: 3

0	1	2	3	4	5	6	7	8	9	10	11	12
		5	1	10			24		3	12	29	17

11. Operation: **insert**(15) mit opt. Position: 6

0	1	2	3	4	5	6	7	8	9	10	11	12
		5	1	10		15	24		3	12	29	17

12. Operation: **delete**(10) mit opt. Position: 4

0	1	2	3	4	5	6	7	8	9	10	11	12
		5	1			15	24		3	12	29	17

13. Operation: **insert**(14) mit opt. Position: 3

0	1	2	3	4	5	6	7	8	9	10	11	12
		5	1	14		15	24		3	12	29	17

14. Operation: **delete**(1) mit opt. Position: 3

Aufgabe 6.3 – Linear Probing

1. Operation: **insert**(16) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
									16			

2. Operation: **insert**(3) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3		

3. Operation: **insert**(12) mit opt. Position: 10

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3	12	

4. Operation: **insert**(17) mit opt. Position: 12

0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3	12	17

5. Operation: **insert**(29) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
29									16	3	12	17

6. Operation: **insert**(10) mit opt. Position: 4

0	1	2	3	4	5	6	7	8	9	10	11	12
29				10					16	3	12	17

7. Operation: **insert**(24) mit opt. Position: 7

0	1	2	3	4	5	6	7	8	9	10	11	12
29				10			24		16	3	12	17

8. Operation: **delete**(16) mit opt. Position: 9

0	1	2	3	4	5	6	7	8	9	10	11	12
				10			24		3	12	29	17

9. Operation: **insert**(5) mit opt. Position: 2

0	1	2	3	4	5	6	7	8	9	10	11	12
		5		10			24		3	12	29	17

10. Operation: **insert**(1) mit opt. Position: 3

0	1	2	3	4	5	6	7	8	9	10	11	12
		5	1	10			24		3	12	29	17

11. Operation: **insert**(15) mit opt. Position: 6

0	1	2	3	4	5	6	7	8	9	10	11	12
		5	1	10		15	24		3	12	29	17

12. Operation: **delete**(10) mit opt. Position: 4

0	1	2	3	4	5	6	7	8	9	10	11	12
		5	1			15	24		3	12	29	17

13. Operation: **insert**(14) mit opt. Position: 3

0	1	2	3	4	5	6	7	8	9	10	11	12
		5	1	14		15	24		3	12	29	17

14. Operation: **delete**(1) mit opt. Position: 3

0	1	2	3	4	5	6	7	8	9	10	11	12
		5	14			15	24		3	12	29	17

Aufgabe 6.4 – Double Hashing

Doppel-Hashing ist eine Methode zur Kollisionsbehandlung. Bei Kollisionen kommt eine Sondierungsfunktion zum Einsatz, die eine sekundäre Hashfunktion beinhaltet:

$$s(x, i) = i \cdot h_2(x), i \in \mathbb{N}_0$$

Diese Sondierungsfunktion wird angewendet, falls der durch die primäre Hashfunktion $h_1(x)$ berechnete Index bereits besetzt ist. Dabei wird i beginnend bei 0 bei jedem Versuch um 1 erhöht. Die vollständige Hashfunktion lautet dann:

$$h(x, i) = (h_1(x) + s(x, i)) \bmod m$$

Verwenden Sie im Folgenden die Hashfunktionen

$$\begin{aligned} h_1(x) &= (3x + 1) \bmod m \\ h_2(x) &= 1 + (x \bmod (m - 1)) \end{aligned}$$

- a) Geben Sie die vollständige Hashfunktion $h(x, i)$ für eine Tabelle der Länge $m = 13$ an.

$$h(x, i) = (3x + 1) \bmod m + i \cdot (1 + (x \bmod (m-1))) \bmod m$$

- b) Veranschaulichen Sie schrittweise das Einfügen der Schlüssel 12, 23, 13, 56, 26, 45, 24, 94, 42 in eine Hashtabelle der Länge $m = 13$. Nutzen Sie hierfür den folgenden Vordruck:

Aufgabe 6.4 – Double Hashing

a) $h(x, i) = ((3x + 1) \bmod 13 + i \cdot (1 + (x \bmod 12))) \bmod 13$

	ins(12)	ins(23)	ins(13)	ins(56)	ins(26)	ins(45)	ins(24)	ins(94)	ins(42)
0:									
1:									
2:									
3:									
4:									
5:									
6:									
7:									
8:									
9:									
10:									
11:									
12:									

Aufgabe 6.4 – Double Hashing

a) $h(x, i) = ((3x + 1) \bmod 13 + i \cdot (1 + (x \bmod 12))) \bmod 13$

	ins(12)
0:	
1:	
2:	
3:	
4:	
5:	
6:	
7:	
8:	
9:	
10:	
11:	12
12:	

Insert(23)

Aufgabe 6.4 – Double Hashing

a) $h(x, i) = ((3x + 1) \bmod 13 + i \cdot (1 + (x \bmod 12))) \bmod 13$

	ins(12)	ins(23)
0:		
1:		
2:		
3:		
4:		
5:		23
6:		
7:		
8:		
9:		
10:		
11:	12	12
12:		

Insert(13)

Aufgabe 6.4 – Double Hashing

a) $h(x, i) = ((3x + 1) \bmod 13 + i \cdot (1 + (x \bmod 12))) \bmod 13$

	ins(12)	ins(23)	ins(13)
0:			
1:			13
2:			
3:			
4:			
5:		23	23
6:			
7:			
8:			
9:			
10:			
11:	12	12	12
12:			

Insert(56)

Aufgabe 6.4 – Double Hashing

a) $h(x, i) = ((3x + 1) \bmod 13 + i \cdot (1 + (x \bmod 12))) \bmod 13$

	ins(12)	ins(23)	ins(13)	ins(56)
0:				56
1:			13	13
2:				
3:				
4:				
5:		23	23	23
6:				
7:				
8:				
9:				
10:				
11:	12	12	12	12
12:				

Insert(26)

Aufgabe 6.4 – Double Hashing

a) $h(x, i) = ((3x + 1) \bmod 13 + i \cdot (1 + (x \bmod 12))) \bmod 13$

	ins(12)	ins(23)	ins(13)	ins(56)	ins(26)
0:				56	56
1:			13	13	13
2:					
3:					
4:					26
5:		23	23	23	23
6:					
7:					
8:					
9:					
10:					
11:	12	12	12	12	12
12:					

Insert(45)

Aufgabe 6.4 – Double Hashing

a) $h(x, i) = ((3x + 1) \bmod 13 + i \cdot (1 + (x \bmod 12))) \bmod 13$

	ins(12)	ins(23)	ins(13)	ins(56)	ins(26)	ins(45)
0:				56	56	56
1:			13	13	13	13
2:						
3:						
4:					26	26
5:		23	23	23	23	23
6:						45
7:						
8:						
9:						
10:						
11:	12	12	12	12	12	12
12:						

Insert(24)

Aufgabe 6.4 – Double Hashing

a) $h(x, i) = ((3x + 1) \bmod 13 + i \cdot (1 + (x \bmod 12))) \bmod 13$

	ins(12)	ins(23)	ins(13)	ins(56)	ins(26)	ins(45)	ins(24)
0:				56	56	56	56
1:			13	13	13	13	13
2:							
3:							
4:					26	26	26
5:		23	23	23	23	23	23
6:						45	45
7:							
8:							24
9:							
10:							
11:	12	12	12	12	12	12	12
12:							

Insert(94)

Aufgabe 6.4 – Double Hashing

a) $h(x, i) = ((3x + 1) \bmod 13 + i \cdot (1 + (x \bmod 12))) \bmod 13$

	ins(12)	ins(23)	ins(13)	ins(56)	ins(26)	ins(45)	ins(24)	ins(94)
0:				56	56	56	56	56
1:			13	13	13	13	13	13
2:								
3:								
4:					26	26	26	26
5:		23	23	23	23	23	23	23
6:						45	45	45
7:								
8:							24	24
9:								
10:								94
11:	12	12	12	12	12	12	12	12
12:								

Insert(42)

Aufgabe 6.4 – Double Hashing

a) $h(x, i) = ((3x + 1) \bmod 13 + i \cdot (1 + (x \bmod 12))) \bmod 13$

	ins(12)	ins(23)	ins(13)	ins(56)	ins(26)	ins(45)	ins(24)	ins(94)	ins(42)
0:				56	56	56	56	56	56
1:			13	13	13	13	13	13	13
2:									
3:									
4:					26	26	26	26	26
5:		23	23	23	23	23	23	23	23
6:						45	45	45	45
7:									
8:							24	24	24
9:									
10:								94	94
11:	12	12	12	12	12	12	12	12	12
12:									42