Tutorium Grundlagen: Algorithmen und Datenstrukturen

Übungsblatt Woche 6

Die Namen {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 g_2: Adalfuns \mapsto 1 Berahthraban \mapsto 4 Chlothar \mapsto 3 Dagoberta \mapsto 2 Egino \mapsto 3 g_3: Adalfuns \mapsto 4 Berahthraban \mapsto 4 Chlothar \mapsto 1 Dagoberta \mapsto 4 Egino \mapsto 2 g_4: Adalfuns \mapsto 3 Berahthraban \mapsto 4 Chlothar \mapsto 1 Dagoberta \mapsto 4 Egino \mapsto 3 g_5: Adalfuns \mapsto 4 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
```

```
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 3 Egino \mapsto 3 g_5: Adalfuns \mapsto 4 Berahthraban \mapsto 2 Chlothar \mapsto 2 Dagoberta \mapsto 2 Egino \mapsto 3
```

Paar	$\int f_1$	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									

```
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_2: Adalfuns \mapsto 1 Berahthraban \mapsto 4 Chlothar \mapsto 3 Dagoberta \mapsto 2 Egino \mapsto 3 g_3: Adalfuns \mapsto 4 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 3 Egino \mapsto 3
```

Paar	$\int f_1$	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	

Paar	$\int f_1$	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	

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

Paar	$\int f_1$	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.

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

Konstruieren Sie eine statische perfekte Hashtabelle für die Elemente:

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

$$(\sum_{i=0}^{2} 2^{i}x_{i}) \mod 17$$

$$(\sum_{i=0}^{2} a_{i}x_{i}) \mod 7 \text{ mit } \mathbf{a} = (0,0,1) \text{ oder } \mathbf{a} = (6,6,2)$$

$$(\sum_{i=0}^{2} a_{i}x_{i}) \mod 3 \text{ mit } \mathbf{a} = (1,0,0) \text{ oder } \mathbf{a} = (0,2,2).$$

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

$$(\sum_{i=0}^{2} 2^{i}x_{i}) \mod 17$$

$$(\sum_{i=0}^{2} a_{i}x_{i}) \mod 7 \text{ mit } \mathbf{a} = (0,0,1) \text{ oder } \mathbf{a} = (6,6,2)$$

$$(\sum_{i=0}^{2} a_{i}x_{i}) \mod 3 \text{ mit } \mathbf{a} = (1,0,0) \text{ oder } \mathbf{a} = (0,2,2).$$

Stufe 1: Stufe 2:

```
(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) \left(\sum_{i=0}^{2} 2^{i}x_{i}\right) \bmod 17
\left(\sum_{i=0}^{2} a_{i}x_{i}\right) \bmod 7 \ \text{mit } \mathbf{a} = (0, 0, 1) \ \text{oder } \mathbf{a} = (6, 6, 2)
\left(\sum_{i=0}^{2} a_{i}x_{i}\right) \bmod 3 \ \text{mit } \mathbf{a} = (1, 0, 0) \ \text{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

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

```
\begin{array}{ll} \text{insert} & 16, 3, 12, 17, 29, 10, 24 \\ \text{delete} & 16 \\ \text{insert} & 5, 1, 15 \\ \text{delete} & 10 \\ \text{insert} & 14 \\ \text{delete} & 1 \\ \end{array}
```

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 $i, (i + 1) \mod m, (i + 2) \mod m, \ldots, j$ 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

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

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

0	1	2	3	$\mid 4$ $\stackrel{\cdot}{}$	\int	6	7	8	9	10	11	12
									16			

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

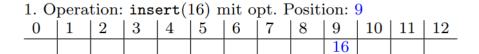
1. Operation: insert(16) mit opt. Position: 9

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

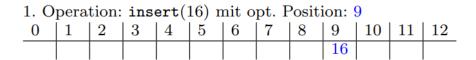
2. Operation: insert(3) mit opt. Position: 9

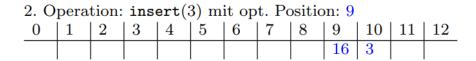
0	1	2	3	$\mid 4$ $\stackrel{`}{}$	$\stackrel{'}{5}$	6	7	8	9	10	11	12
									16	3		

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



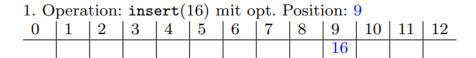
4. Operation: insert(17) mit opt. Position: 12

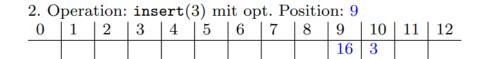


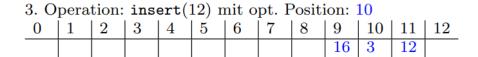


3. O	pera	tion:	ins	$\mathtt{ert}(1)$	12) n	nit o	pt. F	Positi	on:	10		
0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3	12	

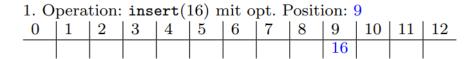
5. Operation: insert(29) mit opt. Position: 9

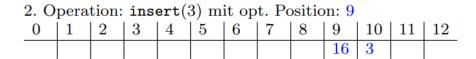


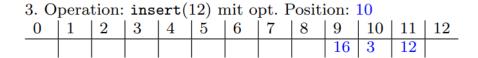




6. Operation: insert(10) mit opt. Position: 4







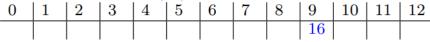
4. O	pera	tion:	ins	$\mathtt{ert}(1)$	17) n	nit o	pt. F	Positi	on:	12		
0	1	2	3	4	5	6	7	8	9	10	11	12
									16	3	12	17

5. O	pera	tion:	ins	ert(:	29) n	nit o	pt. F	Positi	ion:	9		
0	1	2	3	4	5	6						
29									16	3	12	17

6. O	pera	tion:	ins	ert(i	10) n	nit o	pt. F	ositi	ion: 4	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

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



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

0	1	2	3	4	$\overline{5}$	6	7	8	9	10	11	12
									16	3		

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

0	1	2	3	$\mid 4 \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $	$\stackrel{\frown}{5}$	6	7	8	9	10	11	12
									16	3	12	

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

1		,	_	8	9			
					16	3	12	17

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

0	1	2	3	$\mid 4 \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $	$\stackrel{\frown}{5}$	6	7				12
29								16	3	12	17

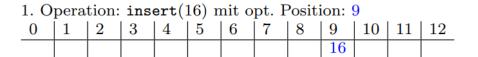
6. Operation: insert(10) mit opt. Position: 4

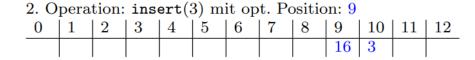
0	1	2	3	4	5	6	7				
29				10				16	3	12	17

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

0	1	2	3	$\mid 4 \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $	$\stackrel{\frown}{5}$					12
29				10		24	16	3	12	17

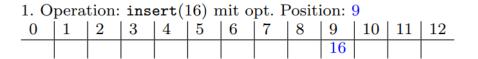
8. Operation: delete(16) mit opt. Position: 9

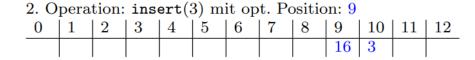


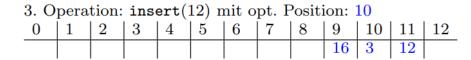


8. O	pera	tion:	del	${\sf ete}(1$	16) n	nit o	pt. F	Positi	on:	9		
0	1	2	3	4							11	
				10			24		3	12	29	17

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







```
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
```

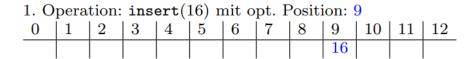
```
9. Operation: insert(5) mit opt. Position: 2

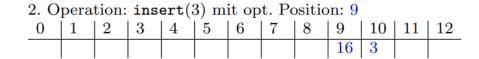
0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12

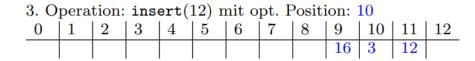
| 5 | 10 | 24 | 3 | 12 | 29 | 17
```

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

17.05.2023 Tobias Eppacher 21







```
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
```

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

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

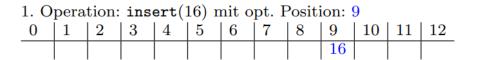
| 5 | 10 | 24 | 3 | 12 | 29 | 17
```

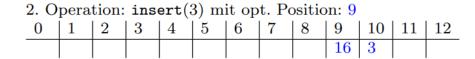
```
10. Operation: insert(1) mit opt. Position: 3

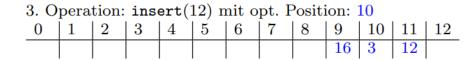
0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12

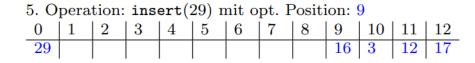
| 5 | 1 | 10 | | 24 | 3 | 12 | 29 | 17
```

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









```
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
```

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

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

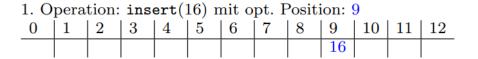
| 5 | 10 | 24 | 3 | 12 | 29 | 17
```

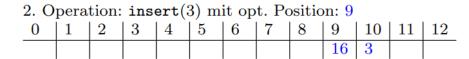
```
10. Operation: insert(1) mit opt. Position: 3

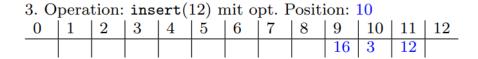
0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12

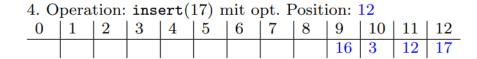
| 5 | 1 | 10 | | 24 | 3 | 12 | 29 | 17
```

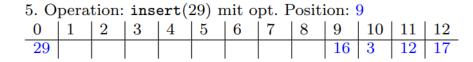
12. Operation: delete(10) mit opt. Position: 4

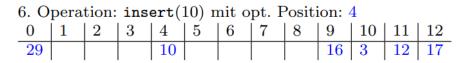












7. Operation:
$$insert(24)$$
 mit opt. Position: 7
0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12
29 | | | 10 | | 24 | | 16 | 3 | 12 | 17

```
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

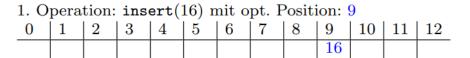
| 5 | 1 | 10 | | 24 | 3 | 12 | 29 | 17
```

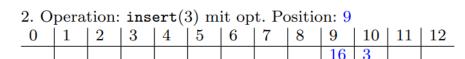
```
12. Operation: delete(10) mit opt. Position: 4

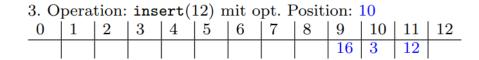
0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12

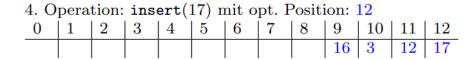
| 5 | 1 | | 15 | 24 | 3 | 12 | 29 | 17
```

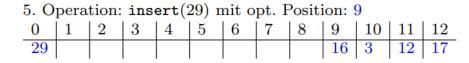
13. Operation: insert(14) mit opt. Position: 3

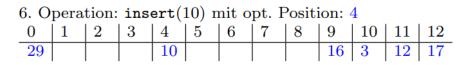




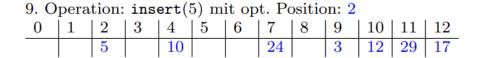








7. Operation:
$$insert(24)$$
 mit opt. Position: 7
0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12
29 | | | 10 | | 24 | | 16 | 3 | 12 | 17

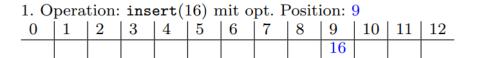


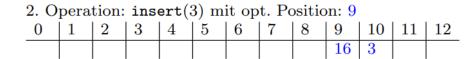
```
12. Operation: delete(10) mit opt. Position: 4

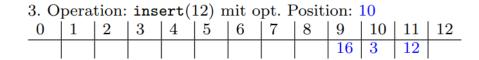
0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12

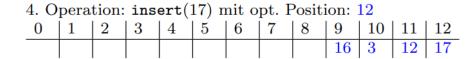
| 5 | 1 | | 15 | 24 | 3 | 12 | 29 | 17
```

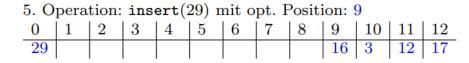
14. Operation: delete(1) mit opt. Position: 3

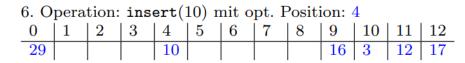


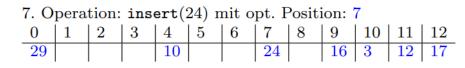


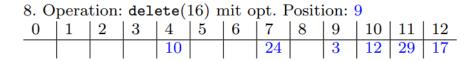


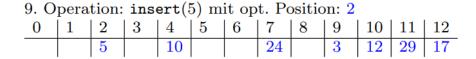


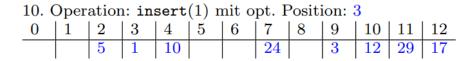


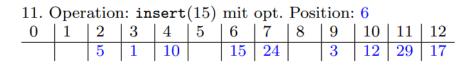


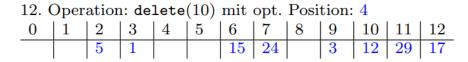












Doppel-Hashing ist eine Methode zur Kollisionsbehandlung. Bei Kollisionen kommt eine Sondierungsfunktion zum Einsatz, die eine sekundäre Hashfunktion beinhaltet:

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

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

$$h_1(x) = (3x+1) \mod m$$

 $h_2(x) = 1 + (x \mod (m-1))$

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

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

	ins(12)	ins(23)	$\mathtt{ins}(13)$	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:									

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

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

Insert(23)

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

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

Insert(13)

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

	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)

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

	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)

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

	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)

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

	ins(12)	$\mathtt{ins}(23)$	$\mathtt{ins}(13)$	ins(56)	ins(26)	$\mathtt{ins}(45)$
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)

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

	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)

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

	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)

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

	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