Tutorium Grundlagen: Algorithmen und Datenstrukturen

Übungsblatt Woche 10

Aufgabe 10.1 Universelles Hashing

Die Pinguingattungen {Brillenpinguin, Zwergpinguin, Eselspinguin, Kaiserpinguin, Goldschopfpinguin} sollen in einer Hash-Tabelle der Größe m = 4 untergebracht werden. Es seien folgende Hashfunktionen gegeben:

In der Vorlesung haben wir den Begriff der c-universellen Hashfunktionen kennengelernt.

- a) Geben Sie für die Familie \mathcal{H}_1 = {f1, f2, f3, f4} das kleinste c an, so dass H_1 c-universell ist.
- b) Finden Sie eine möglichst kleine Familie $\mathcal{H}_2 \subseteq \{g1, g2, g3, g4, g5\}$, die 1-universell ist. Untermauern Sie Ihre Aussagen mit glaubwürdigen Argumenten.

Aufgabe 10.1 (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.

$$\frac{|\{f \in \mathcal{H} : f(x) = f(y)\}|}{|\mathcal{H}|} \le \frac{c}{m} \qquad \forall x \ne y$$

Paar	$\int f_1$	f_2	f_3	f_4	g_1	g_2	g_3	g_4	g_5
Brillenpinguin/Zwergpinguin			X		X		X		
Brillenpinguin/Eselspinguin						X			
Brillenpinguin/Kaiserpinguin		X					X	X	
Brillenpinguin/Goldschopfpinguin	X							X	
Zwergpinguin/Eselspinguin	X			X					X
Zwergpinguin/Kaiserpinguin							X		X
Zwergpinguin/Goldschopfpinguin		X				X			
Eselspinguin/Kaiserpinguin									X
Eselspinguin/Goldschopfpinguin					X				
Kaiserpinguin/Goldschopfpinguin			X	X				X	

Aufgabe 10.1 (a)

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.

$$\frac{|\{f \in \mathcal{H} : f(x) = f(y)\}|}{|\mathcal{H}|} \le \frac{c}{m} \qquad \forall x \ne y$$

Paar	f_1	f_2	f_3	f_4	g_1	g_2	g_3	g_4	g_5
Brillenpinguin/Zwergpinguin			X		X		X		
Brillenpinguin/Eselspinguin						X			
Brillenpinguin/Kaiserpinguin		X					X	X	
Brillenpinguin/Goldschopfpinguin	X							X	
Zwergpinguin/Eselspinguin	X			X					X
Zwergpinguin/Kaiserpinguin							X		X
Zwergpinguin/Goldschopfpinguin		X				X			
Eselspinguin/Kaiserpinguin									X
Eselspinguin/Goldschopfpinguin					X				
Kaiserpinguin/Goldschopfpinguin			X	X				X	

Aufgabe 10.2 Die perfekte Hashtabelle

Konstruieren Sie eine statische perfekte Hashtabelle für die Elemente:

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

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

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

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

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

Erinnerung: perfektes statisches Hashing

- Ziehe Hashfunktion aus einer c-universellen Familie $H_{\lceil\sqrt{2}cn
 ceil}$ bis Kollisionszahl kleiner gleich $\sqrt{2}n$
- Jedes Bucket hat dann Größe $m_l=cb_l(b_l-1)+1$, wobei b_l die Elementzahl im jeweiligen Bucket ist
- Ziehe für jedes Bucket aus einer c-universellen Familie H_{m_l} bis im Bucket alle Schlüssel injektiv abgebildet werden

In dieser Aufgabe sind die möglichen Hashfunktionen schon gegeben

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

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

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

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

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

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

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

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

Tobias Eppacher

Aufgabe 10.3 Linear Probing

Veranschaulichen Sie Hashing mit Linear Probing. Die Größe der Hashtabelle ist dabei jeweils m = 13. Führen Sie die folgenden Operationen aus:

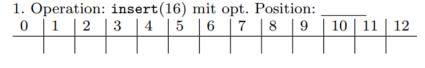
```
\begin{array}{ll} \text{insert} & 16, 3, 12, 17, 29, 10, 24 \\ \text{delete} & 16 \\ \text{insert} & 5, 1, 15 \\ \text{delete} & 10 \\ \text{insert} & 14 \\ \text{delete} & 1 \\ \end{array}
```

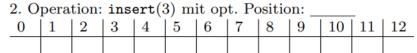
Verwenden Sie die Hashfunktion:

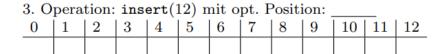
$$h(x) = 3x \mod 13$$

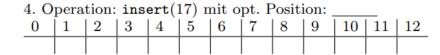
Beim Löschen soll die dritte Methode aus der Vorlesung verwendet werden, d.h. die Wiederherstellung der folgenden Invariante:

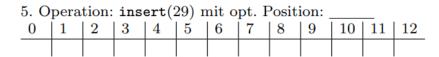
Für jedes Element e in der Hashtabelle mit Schlüssel k(e), aktueller Position j und optimaler Position i = h(k(e)) sind alle Positionen i, $(i + 1) \mod m$, $(i + 2) \mod m$, ..., j der Hashtabelle belegt. Bei dieser Aufgabe soll keine dynamische Größenanpassung der Hashtabelle stattfinden.

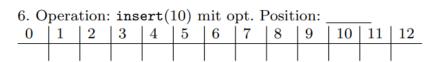


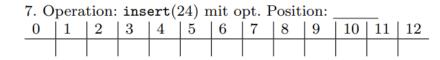


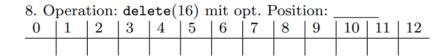


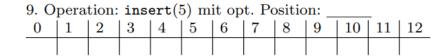


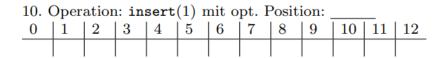


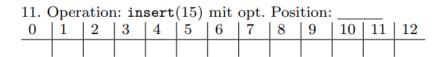


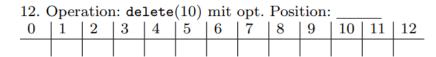


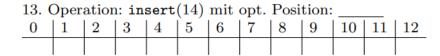


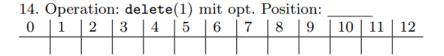


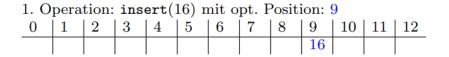


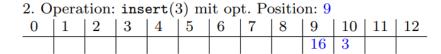


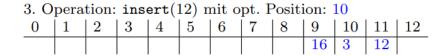


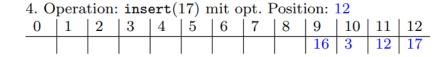


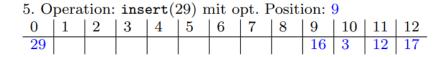


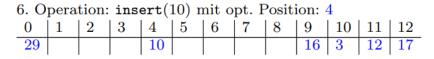


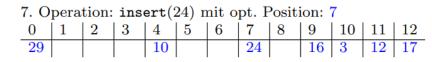




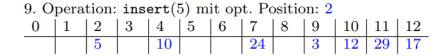




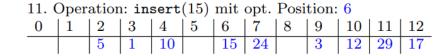




8. Operation: delete(16) mit opt. Position: 9												
0	1	2	3	4	5							
				10			24		3	12	29	17



10. (10. Operation: insert(1) mit opt. Position: 3													
0	1	2	3	4										
		5	1	10			24		3	12	29	17		



12. Operation: delete(10) mit opt. Position: 4												
0	1											
		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	

Aufgabe 10.4 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

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

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

- 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. (Siehe Vordruck)

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

	ins(12)	ins(23)	ins(13)	ins(56)	ins(26)	ins(45)	ins(24)	ins(94)	lns(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