Grundlagen: Algorithmen und Datenstrukturen

Woche 11

Tobias Eppacher

School of Computation, Information and Technology

7. Juli 2025

Inhalt

Aufgabe 11.1 - 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:

f_1 :	Brillenp. $\mapsto$ 4	Zwergp. $\mapsto$ 2	Eselsp. $\mapsto$ 2	Kaiserp. $\mapsto$ 1	Goldschopfp. $\mapsto$ 4
f_2 :	Brillenp. $\mapsto$ 3	Zwergp. $\mapsto$ 4	Eselsp. $\mapsto$ 2	Kaiserp. $\mapsto$ 3	Goldschopfp. $\mapsto$ 4
f_3 :	Brillenp. $\mapsto$ 2	Zwergp. $\mapsto$ 2	Eselsp. $\mapsto$ 4	Kaiserp. $\mapsto$ 1	Goldschopfp. $\mapsto$ 1
f_4 :	Brillenp. $\mapsto$ 1	Zwergp. $\mapsto$ 3	Eselsp. $\mapsto$ 3	Kaiserp. $\mapsto$ 4	Goldschopfp. $\mapsto$ 4
g_1 :	Brillenp. $\mapsto$ 1	Zwergp. $\mapsto$ 1	Eselsp. $\mapsto$ 3	Kaiserp. $\mapsto$ 2	Goldschopfp. $\mapsto$ 3
g_2 :	Brillenp. $\mapsto$ 2	Zwergp. $\mapsto$ 4	Eselsp. $\mapsto$ 2	Kaiserp. $\mapsto$ 3	Goldschopfp. $\mapsto$ 4
g_3 :	Brillenp. $\mapsto$ 4	Zwergp. $\mapsto$ 4	Eselsp. $\mapsto$ 1	Kaiserp. $\mapsto$ 4	Goldschopfp. $\mapsto$ 2
g_4 :	Brillenp. $\mapsto$ 3	Zwergp. $\mapsto$ 1	Eselsp. $\mapsto$ 2	Kaiserp. $\mapsto$ 3	Goldschopfp. $\mapsto$ 3
g_5 :	Brillenp. $\mapsto$ 4	Zwergp. $\mapsto$ 2	Eselsp. $\mapsto$ 2	Kaiserp. $\mapsto$ 2	Goldschopfp. $\mapsto$ 3

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

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 11.1 - Universelles Hashing (a)

Eine Familie von Hashfunktionen $\mathcal{H}$ ist c-universell, wenn für alle Wertepaare x, y mit $x \neq y$ gilt:

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

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

Aufgabe 11.1 - Universelles Hashing (a)

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

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	

Wir suchen kleinst-mögliches c für $\mathcal{H}_1 = \{f_1, f_2, f_3, f_4\}$

Aufgabe 11.1 - Universelles Hashing (a) - Extraplatz

Aufgabe 11.1 - Universelles Hashing (b)

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

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	

Wir suchen eine möglichst kleine Familie $\mathcal{H}_2 \subseteq \{g_1, f_2, g_3, g_4, g_5\}$, sodass $c = 1$ ist.

Aufgabe 11.1 - Universelles Hashing (b) - Extraplatz

Aufgabe 11.2 - Die perfektes 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:

$$\left(\sum_{i=0}^{2} 2^i x_i\right) \mod 17$$

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

$$\left(\sum_{i=0}^{2} a_i x_i\right) \mod 3 \text{ mit } a = (1, 0, 0) \text{ oder } a = (0, 2, 2)$$

Stufe 1: Ziehe aus c-universeller Familie $H_{\lceil\sqrt{2}cn\rceil}$ bis $C(n) < \sqrt{2}n$

Stufe 2: Bei b_ℓ Elementen wähle Bucketgröße $m_\ell = cb_\ell(b_\ell - 1) + 1$. Ziehe aus c-universeller Familie H_{m_ℓ} bis injektiver Abbildung.

Wir müssen hier nicht ziehen, sondern nur die angegebenen Hashfunktionen verwenden.

Aufgabe 11.2 - Die perfektes Hashtabelle

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

Aufgabe 11.2 - Die perfektes Hashtabelle

Aufgabe 11.2 - Die perfektes Hashtabelle

Finale Platzierungen der Elemente in der Hashtabelle:

Element	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

Aufgabe 11.3 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.

Aufgabe 11.3 Linear Probing

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

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

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

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

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

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

Aufgabe 11.3 Linear Probing

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

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

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

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

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

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

Aufgabe 11.3 Linear Probing

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

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

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

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

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

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

Aufgabe 11.3 Linear Probing

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

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

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

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

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

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

Aufgabe 11.3 Linear Probing

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

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

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

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

Aufgabe 11.4 - 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 primare 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:

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

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 $m = 13$.

Aufgabe 11.4 - Double Hashing (a)

$$h(x, i) = (h_1(x) + s(x, i)) \mod m$$
$$s(x, i) = i \cdot h_2(x), i \in \mathbb{N}_0$$

$$h_1(x) = (3x + 1) \mod m$$
$$h_2(x) = 1 + (x \mod (m - 1))$$

Mit $m = 13$

Aufgabe 11.4 - Double Hashing (b)

	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)													
ins(42)													

E-Aufgaben

- Aufgabe 11.5 - Bad Hash
  - Intuition zu Hashfunktion
- Aufgabe 11.6 - Linear Probing
  - Abgeänderte Aufgabe aus Tutorium

Hausaufgaben

- Hausaufgabe 9 - AB-Baum
  (Deadline: 09.07.2025)
- Hausaufgabe 10 - Simple Hashing with Chaining
  (Deadline: 16.07.2025)
- Hausaufgabe 11 - Double Hashing
  (Deadline: 23.07.2025)

Fragen?

- Nach Übung gerne bei mir melden
- Tutoriumschannel oder DM an mich auf Zulip
- Vorlesungschannels von GAD auf Zulip (insbesondere bei Hausaufgaben)

Feedback oder Verbesserungsvorschläge?
Gerne nach dem Tutorium mit mir quatschen oder DM auf Zulip

Bis nächste Woche!