Assignment 7
Advanced Algorithms & Data Structures PS

Christian Müller 1123410
Daniel Kocher, 0926293

May 18, 2016

Aufgabe 14

Sei $U = \{0, \ldots, 30\}$ und $S = \{2, 4, 7, 11, 12, 18, 19, 22, 26, 29\}$. Benutzen Sie das zweistufige Schema aus der Vorlesung um eine perfekte Hash-Funktion mit $k = 3$, $N = 31$ und $n = 10$ zu finden. Berechnen Sie für $i = 0, \ldots, n-1$ die Werte W_i, b_i, m_i, k_i und die Funktion h_{k_i}. Geben Sie anschließend für jedes Element aus S die Position dieses Elementes in der Hash-Tafel an.

Die Hash-Funktion für die erste Stufe wird allgemein wie folgt berechnet:

$$h_k(x) = \big((k \cdot x) \mod N\big) \mod m \tag{1}$$

wobei $m = n = |S| = 10$ ist.

Setzt man nun die Werte aus der Angabe ein - $k = 3$, $N = 31$ und $n = 10$ - bekommt man die folgende konkrete Hash-Funktion für die erste Stufe:

$$h_3(x) = \big((3 \cdot x) \mod 31\big) \mod 10 \tag{2}$$

Nun wird für $i = 0, \ldots, n-1 = 9$

1. $W_i = \big\{x \in S : h_k(x) = i\big\}$ mit $h_k = h_3$
2. $b_i = |W_i|$
3. $m_i = 2 \cdot b_i \cdot (b_i - 1) + 1$

berechnet, siehe Table 1.

i	W_i	b_i	m_i
0		0	1
1	7	1	1
2	4, 11	2	5
3	18	1	1
4	22	1	1
5	12, 29	2	5
6	2, 19, 26	3	13
7		0	1
8		0	1
9		0	1

Table 1: Werte für Schritt 2 des zweistufigen Schemas aus der Vorlesung.

i	s_i
0	0
1	1
2	2
3	7
4	8
5	9
6	14
7	27
8	28
9	29

Table 2: Werte für Schritt 3 des zweistufigen Schemas aus der Vorlesung.

Nun müssen k_i so gewählt werden, dass

$$h_{k_i} : x \longrightarrow \big((k_i \cdot x) \mod N\big) \mod m_i \tag{3}$$

eingeschränkt auf W_i injektiv ist. Durch diese zweite Stufe werden jene Einträge W_i mit $b_i = |W_i| > 1$ auf mehrere "Unterbuckets" abgebildet.
$b_i > 1$ ist für $i = \{2, 5, 6\}$ der Fall. Für $i = \{1, 3, 4\}$ benötigen wir keine zusätzliche Hash-Funktion h_{k_i}, da nur ein einziger Wert im Bucket vorhanden ist - eine weitere Aufteilung auf "Unterbuckets" macht also keinen Sinn.
Wir berechnen nun k_i nach Gleichung 3 für $(i, W_i, m_i) \in \Big\{\big(2, \{4, 11\}, 5\big), \big(5, \{12, 29\}, 5\big), \big(6, \{2, 19, 26\}, 13\big)\Big\}$ und $N = 31$:

- $i = 2$, $W_2 = \{4, 11\}$ und $m_2 = 5$:
  Wähle $k_2 = 1$: Ist h_{k_2} eingeschränkt auf W_2 injektiv?
  $h_{k_2}(4) = 4$ und $h_{k_2}(11) = 1 \Rightarrow h_{k_2}$ ist eingeschränkt auf W_2 injektiv.

- $i = 5$, $W_5 = \{12, 29\}$ und $m_5 = 5$:
  Wähle $k_5 = 1$: Ist h_{k_5} eingeschränkt auf W_5 injektiv?
  $h_{k_5}(12) = 2$ und $h_{k_5}(29) = 4 \Rightarrow h_{k_5}$ ist eingeschränkt auf W_5 injektiv.

- $i = 6$, $W_6 = \{2, 19, 26\}$ und $m_6 = 13$:
  Wähle $k_6 = 1$: Ist h_{k_6} eingeschränkt auf W_6 injektiv?
  $h_{k_6}(2) = 2$, $h_{k_6}(19) = 6$ und $h_{k_6}(26) = 0 \Rightarrow h_{k_6}$ ist eingeschränkt auf W_6 injektiv.

Daraus ergeben sich für die zweite Stufe folgende Hash-Funktionen für $i \in \{2, 5, 6\}$

$$h_{k_2}(x) = h_{k_5}(x) = (x \bmod 31) \mod 5 \tag{4}$$

$$h_{k_6}(x) = (x \bmod 31) \mod 13 \tag{5}$$

und für $i \in \{0, 1, 3, 4, 7, 8, 9\}$

$$h_{k_0}(x) = h_{k_1}(x) = h_{k_3}(x) = h_{k_4}(x) = h_{k_7}(x) = h_{k_8}(x) = h_{k_9}(x) = (x \bmod 31) \mod 1 \tag{6}$$

Basierend auf diesen Hash-Funktionen h_{k_0} bis h_{k_9} wird nun der finale Schritt vorgenommen. Zuerst werden $s_i = \sum_{j<i} m_j$ berechnet und dann wird $x \in S$ in Tafelposition $T[s_i + j]$ gespeichert, wobei $i = h_k(x)$ (erste Hash-Funktion, siehe Gleichung 2) und $j = h_{k_i}(x)$ (zweite Hash-Funktion, siehe Gleichungen 4 - 6).
Für die Berechnung von s_i für $i = \{0, \ldots, 9\}$ siehe Table 2. Die resultierenden Positionen in der Hash-Tafel sind in Table 3 aufgelistet.

x	$s_i + j$ (Position von x)
2	$14 + 2 = \mathbf{16}$
4	$2 + 4 = \mathbf{6}$
7	$1 + 0 = \mathbf{1}$
11	$2 + 1 = \mathbf{3}$
12	$9 + 2 = \mathbf{11}$
18	$7 + 0 = \mathbf{7}$
19	$14 + 6 = \mathbf{20}$
22	$8 + 0 = \mathbf{8}$
26	$14 + 0 = \mathbf{14}$
29	$9 + 4 = \mathbf{13}$

Table 3: Finale Positionen von $x \in S$ in der Hash-Tafel.

Aufgabe 15

Wir nehmen an, dass in einer Datenstruktur die i-te Operation Kosten c_i verursacht, wobei $c_i = i$ für alle i der Form $2^k + 1$ (k ist eine natürliche Zahl) und $c_i = 2$ sonst. Zeigen Sie, dass die amortisierten Kosten einer Operation immer konstant sind.

$$c_i = \begin{cases} i, \text{ falls } i = 2^k + 1 \text{ mit } k \in \mathbb{N} \\ 2, \text{ sonst.} \end{cases}$$

Op_i	c_i
1	2
2	2
3	3
4	2
5	5
6	2
7	2
8	2
9	9
10	2

Table 4: Beispiel für $n = 10$ Operationen

Aggregat-Methode mit n Operationen
O. B. d. A. betrachten wir den Fall $n = 2^k$ mit $k \in \mathbb{N}$

1. Gesamtkosten C_1 der Menge (A_1) der Operationen (a_i) für die $i = 2^j + 1$ mit $j \in \mathbb{N}$ und $i \leq n$ gilt:
   Für alle a_i aus A_1 gilt $0 \leq j \leq k - 1$:
   $$C_1 = (2^0 + 1) + (2^1 + 1) + (2^2 + 1) + ... + (2^{k-1} + 1) \tag{7}$$
   wegen $n = 2^k \implies k = \mathrm{ld}(n)$ gilt:
   $$C_1 = (2^0 + 1) + (2^1 + 1) + (2^2 + 1) + ... + (2^{\mathrm{ld}(n)-1} + 1) \tag{8}$$
   $$C_1 = \mathrm{ld}(n) + \sum_{i=0}^{\mathrm{ld}(n)-1} 2^i \tag{9}$$
   Es gilt: (siehe 18)
   $$\sum_{i=0}^{n} 2^i = 2^{n+1} - 1 \tag{10}$$
   Also:
   $$C_1 = \mathrm{ld}(n) + 2^{\mathrm{ld}(n)} - 1 \tag{11}$$
2. Gesamtkosten C_2 der Menge (A_2) der Operationen (a_i) für die $i \neq 2^j + 1$ mit $j \in \mathbb{N}$ gilt:
   $|A1| = \mathrm{ld}(n)$ und $|A_1| + |A_2| = n \implies |A_2| = n - \mathrm{ld}(n)$
   $$C_2 = 2 \cdot (n - \mathrm{ld}(n)) \tag{12}$$

Seien C die Gesamtkosten der n Operationen dann:

$$C = C_1 + C_2 = \mathrm{ld}(n) + 2^{\mathrm{ld}(n)} - 1 + 2 \cdot (n - \mathrm{ld}(n)) \tag{13}$$
$$C = \mathrm{ld}(n) + n - 1 + 2n - 2\mathrm{ld}(n) \tag{14}$$
$$C = 3n - \mathrm{ld}(n) - 1 \tag{15}$$
$$3n - \mathrm{ld}(n) - 1 \tag{16}$$
$$a_i = \frac{3n - \mathrm{ld}(n) - 1}{n} < 3 \tag{17}$$

Die amortisierten Kosten einer Operation sind höchstens 3 und somit in $O(1)$.

Proof.

$$\sum_{i=0}^{n} 2^i = 2^{n+1} - 1 \tag{18}$$

IB: $n = 0$

$$\sum_{i=0}^{0} 2^i = 2^0 = 1 = 2^1 - 1 = 2^{0+1} - 1 \tag{19}$$

IH: Für ein beliebes aber festes $k \in \mathbb{N}$ gilt:

$$\sum_{i=0}^{k} 2^i = 2^{k+1} - 1 \tag{20}$$

IS: $k \to k + 1$
Zu zeigen:

$$\sum_{i=0}^{k+1} 2^i = 2^{k+2} - 1 \tag{21}$$

$$\begin{aligned}
\sum_{i=0}^{k+1} 2^i &= \sum_{i=0}^{k} 2^i + 2^{k+1} \\
&= 2^{k+1} - 1 + 2^{k+1} \\
&= 2 \cdot 2^{k+1} - 1 \\
&= 2^{k+2} - 1
\end{aligned}$$

□