

Algorithmen und Datenstrukturen
SS 2015

Übungsblatt 10: Lineares Hashing

Besprechung: 6.7.– 10.7.

Abgabe aller mit **Punkten** versehenen Aufgaben bis 5.7.2015 17:35

Aufgabe 10-1 Hashfunktionen für Lineares Hashing

Tutoraufgabe

Lineares Hashing benötigt eine Folge von Hashfunktionen $h_L(k)$ für jedes Level $L \geq 0$, die sowohl die Bereichs- als auch die Splitbedingung erfüllen:

$$h_L(k) \in \{0, \dots, N \cdot 2^L - 1\}$$

$$h_{L+1}(k) = h_L(k) \quad \text{oder} \quad h_{L+1}(k) = h_L(k) + N \cdot 2^L$$

(a) Sei $h_L(k) := k \bmod (N \cdot 2^L)$.

Beweisen Sie, dass diese Folge von Hashfunktionen die Splitbedingung erfüllt.

(b) Sei $\mathcal{H}(k) \rightarrow \mathbb{N}$ eine beliebige Hashfunktion (ohne Bereichs- oder Splitbedingung).

Zeigen Sie: $h_L(k) := \mathcal{H}(k) \bmod (N \cdot 2^L)$ erfüllt die Bereichs- und die Splitbedingung.

(c) Sei p die Position des Expansionszeigers. Welche Seitennummer hat die nächste Seite, die erzeugt wird?

(d) Beim Linearen Hashing wird der Bucket p gesplittet. Zeigen Sie, dass alle Elemente aus Bucket p entweder in p bleiben, oder in den neuen Bucket gespeichert werden (bzw. in Überlaufseiten dieser Buckets).

(e) Sei p die Position des Expansionszeigers. Mit welcher Formel können Sie berechnen, ob für einen Schlüssel k schon die neue – oder noch die alte – Hashfunktion verwendet werden muss?

(f) Nach dem Split einer Seite wird der Expansionszeiger p erhöht, außer die Datei hat sich verdoppelt. In diesem Falle wird der Zeiger auf $p = 0$ gesetzt. Warum?

(g) Die Hashfunktion $h_{L+1}(k)$ hat nach der Bereichsbedingung den Wertebereich $\{0, \dots, N \cdot 2^{L+1} - 1\}$. Angenommen unsere Tabelle hat sich *noch nicht* verdoppelt, und enthält aktuell nur $N \cdot 2^{L+1} - 2$ Seiten. Warum ist es ausgeschlossen, dass wir einen Wert in der nicht existierenden Seite $N \cdot 2^{L+1} - 1$ speichern wollen, obwohl diese im Wertebereich der neuen Hashfunktion liegt?

(h) Wie können Sie die Splitbedingung nutzen, um Fehler beim Bearbeiten der Aufgabe zu erkennen?

Aufgabe 10-2 Lineares Hashing**Tutoraufgabe**

Gegeben sei eine Hashtabelle mit Primärseiten der Größe 2 und Überlaufseiten der Größe 1, die anfangs aus 2 leeren Primärseiten besteht. Die Folge der verwendeten Hashfunktionen sei $h_0, h_1, h_2 \dots$ mit $h_i(x) = x \bmod (2 \cdot 2^i)$. Eine Expansion der Hashtabelle soll immer dann erfolgen, wenn der Belegungsfaktor 0.80 überschritten wird.

Verwenden Sie lineares Hashing und fügen Sie in die Hashtabelle die Schlüssel 7, 5, 13, 19, 34, 3, 30, 21, 2, 14, 15, 22, 27, 29 und 10 ein.

Stellen Sie den Zustand der Hashtabelle (Primärseiten, Überlaufseiten, Expansionszeiger) nach jeder Expansion und nach der letzten Einfügung graphisch dar.

Aufgabe 10-3 Lineares Hashing**6 Punkte**

Gegeben sei eine Hashtabelle mit Primärseiten der Größe 2 und Überlaufseiten der Größe 1, die anfangs aus 2 leeren Primärseiten besteht. Die Folge der verwendeten Hashfunktionen sei $h_0, h_1, h_2 \dots$ mit $h_i(x) = x \bmod (2 \cdot 2^i)$. Eine Expansion der Hashtabelle soll immer dann erfolgen, wenn der Belegungsfaktor 0.80 überschritten wird.

Verwenden Sie lineares Hashing (*ohne* partielle Erweiterungen) und fügen Sie in die Hashtabelle die Schlüssel 3, 2, 15, 10, 25, 26, 11, 7, 38 und 34 ein.

Stellen Sie den Zustand der Hashtabelle (Primärseiten, Überlaufseiten, Expansionszeiger) nach jeder Expansion und nach der letzten Einfügung graphisch dar.

Hinweis: verwenden Sie den Belegungsfaktor. Wenn Sie durch Verwendung des falschen Kriteriums ein mal nicht expandieren müssen, müssen unsere Korrektoren den Rest der Aufgabe mit 0 Punkten werten, um den Korrekturaufwand und die -Zeit vertretbar zu halten.

Kontrollieren Sie wiederholt, dass Sie keine Elemente verloren oder geändert haben. Finden Sie ihren Fehler und korrigieren Sie ihn – auch hier können wir keine “Folgefehler” zählen, da sich die Aufgabe dadurch deutlich vereinfachen kann. Ein nicht erfolgter Split gibt keine Punkte, auch wenn ein vorhergegangener Fehler dafür verantwortlich ist.

Aufgabe 10-4 Hashfunktionen für Lineares Hashing mit partiellen Erweiterungen Tutoraufgabe

Lineares Hashing benötigt eine Folge von Hashfunktionen $h_L(k)$ für jedes Level $L \geq 0$, die die Splitbedingung erfüllen:

$$h_{L+1}(k) = h_L(k) \quad \text{oder} \quad h_{L+1}(k) = h_L(k) + N \cdot 2^L$$

Für die partiellen Erweiterungen können wir diese Splitbedingung wie folgt verallgemeinern:

$$h_{L+1}(k) \bmod M_L = h_L(k)$$

für ein geeignetes M_L in Abhängigkeit vom Level L .

Im Folgenden nehmen wir $n_0 = 2$ an.

- (a) Zeigen Sie, dass die Splitbedingung für Lineares Hashing mit $M_L = N \cdot 2^L$ dieser Bedingung entspricht.
- (b) Sei $M_L = N \cdot 2^L$. Geben Sie die minimale und die maximale Größe der Tabelle in Abhängigkeit von der aktuellen partiellen Erweiterung an.
- (c) Sei $\mathcal{H}(x)$ eine beliebige Hashfunktion für Objekte x .
Zeigen Sie, dass

$$h_L(i, x) := \mathcal{H}(x) \bmod (i+1) \cdot N \cdot 2^L \quad (= \mathcal{H}(x) \bmod (i+1) \cdot M_L)$$

eine geeignete Familie von Hashfunktionen für das lineare Hashing mit partiellen Erweiterungen darstellt. Dabei sei $i \in \{1, \dots, n_0 + 1\}$ abhängig von der partiellen Erweiterung.

Zeigen Sie zunächst, dass $h_L(3, x) = h_{L+1}(1, x)$ gilt.

Anschließend zeigen Sie, dass oben genannte Splitbedingung für $h_L(1, x)$ gilt, aber mit $M'_L = 2M_L$.

Diese Hashfunktionen entsprechen also dem verdoppeln der Hashtabelle.

Die Funktion $h_L(2, x)$ ist dazwischen angesiedelt, und für die partiellen Erweiterungen zuständig.

Zeigen Sie nun, dass $h_L(2, x) \bmod M_L = h_L(1, x) \bmod M_L$ (nicht M'_L von oben!).

Zeigen Sie ebenfalls, dass $h_L(2, x) \bmod M_L = h_L(3, x) \bmod M_L$.

Was bedeutet dieses Ergebnis?

- (d) Gegeben ein Schlüssel x . Wie finden Sie, welche Hashfunktion aktuell zu verwenden ist (sowohl für das Einfügen, als auch für die Suche)?
- (e) Wiederholen Sie die Unterschiede zwischen linearem Hashing mit und ohne partiellen Erweiterungen: Wie ist die initiale Tabellengröße, wie sehen die Hashfunktionen aus, wie sehen Level/Expansionen aus, wie verhält sich der Expansionszeiger, welches Expansionskriterium wird verwendet, und wie werden Seiten aufgespalten?

Aufgabe 10-5 Lineares Hashing mit partiellen Erweiterungen**Tutoraufgabe**

Gegeben sei eine Hashtabelle mit der initiale Größe $2N = 4$ ($N = 2$).
 Primärseiten haben die Kapazität $b = 4$, Überlaufseiten die Kapazität $c = 2$.

Als Strategie verwenden Sie Lineares Hashing mit partiellen Erweiterungen.

Als Hashfunktionen verwenden Sie: $h_L(1, k) := k \bmod (2 \cdot 2^L N)$, $h_L(2, k) := k \bmod (3 \cdot 2^L N)$

Expandieren Sie, wenn die Speicherplatzausnutzung den Schwellwert $\alpha = 0.85$ überschreitet.

Nach den ersten Einfügungen sei die Tabelle wie folgt:

320	757	090	711
016	613	402	027
712	-	522	303
004	-	-	319

Welche Hashfunktionen werden aktuell verwendet, wo befindet sich der Expansionszeiger?

Fügen Sie folgende Elemente zusätzlich in die Tabelle ein:

434, 435, 215, 125, 122, 233, 007, 014, 640, 188, 054, 042 und 420.

Geben Sie an, welche Zellen an der nächsten Expansion beteiligt wären.

Aufgabe 10-6 Lineares Hashing mit partiellen Erweiterungen**9 Punkte**

Verwenden Sie das Verfahren lineares Hashing mit $n_0 = 2$ partiellen Erweiterungen.

Gegeben Sei eine Hashtabelle mit $N = 1$, Primärseiten Größe $b = 2$, Überlaufseiten Größe $c = 1$.

Expandieren Sie, wenn die Speicherplatzausnutzung $\alpha = 0.70$ übersteigt.

Als Hashfunktion verwenden Sie: $h_L(n, k) = k \bmod 2^L(n + 1)N$

Nach dem Einfügen der Schlüssel 54 und 55 sieht die Hashtabelle ($L = 0$) wie folgt aus:

54	55
-	-

Fügen Sie in diese Tabelle zusätzlich folgende Schlüssel (in dieser Reihenfolge) ein:

75, 85, 15, 98, 96, 14, 68, 48, 43, 26, 42 und 65 ein.

Stellen Sie den Zustand der Hashtabelle (Primärseiten, Überlaufseiten) nach jeder Expansion und nach der letzten Einfügung graphisch dar. (Sie *dürfen* in die Tabelle auch schon weitere Einfügungen notieren, *falls* durch diese keine Expansion ausgelöst wird.)

Wichtiger Hinweis: Verwenden Sie die Speicherplatzausnutzung, *keinesfalls* den Belegungsfaktor – expandieren Sie, falls die Kontrollfunktion durch das Einfügen verletzt werden würde, und zeichnen Sie den *neuen* Zustand der Hashtabelle. Wenn Sie die falsche Kontrollfunktion verwenden, müssen die Tutoren die Aufgabe ab dort mit 0 Punkten werten, um den Korrekturaufwand und die -zeit der Aufgabe vertretbar zu halten.)

Kontrollieren Sie wiederholt, dass Sie keine Elemente verloren oder geändert haben. Finden Sie ihren Fehler und korrigieren Sie ihn – auch hier können wir keine “Folgefehler” zählen, da sich die Aufgabe dadurch deutlich vereinfachen kann. Ein nicht erfolgter Split gibt keine Punkte, auch wenn ein vorhergegangener Fehler dafür verantwortlich ist.