

Algorithmen und Datenstrukturen
SS 2013

Übungsblatt 9: Hashing + Sortieralgorithmen

Besprechung: 27.06. - 02.07.2013

Abgabe dieses Übungsblattes bis spätestens Donnerstag, 27.06.2013, 8:00 Uhr.

Hinweise zur Abgabe:

Geben Sie bitte Ihre gesammelten Lösungen zu diesem Übungsblatt in einer Datei `loesung09.zip` unter <https://uniworx.ifi.lmu.de> ab.

Aufgabe 9-1 *Lineares Hashing mit partiellen Erweiterungen*

9 Punkte

Betrachten Sie lineares Hashing mit $n_0 = 2$ partiellen Erweiterungen. Gegeben seien

- $2N$: anfängliche Größe der Hashtabelle (konstant)
- $n \in \{1, \dots, n_0\}$: aktuelle partielle Expansion, anfangs $n = 1$
- $p \in \{0, \dots, N - 1\}$: Expansionszeiger, d.h. nächste zu expandierende Seite, anfangs $p = 0$
- L : Level, d. h. Anzahl der Verdoppelungen der Tabelle seit Beginn, anfangs $L = 0$
- $h_L(n, k)$: Folge von Hashfunktionen
 $h_0(1, k) = k \bmod 2N, h_0(2, k) = k \bmod 3N,$
 $h_1(1, k) = k \bmod 4N, h_1(2, k) = k \bmod 6N,$
 $h_2(1, k) = k \bmod 8N, \dots$

Beachte: Hier ist $n \in \{1, \dots, n_0 + 1\}$ und $h_L(n_0 + 1, k) = h_{L+1}(1, k)$, z.B. $h_0(3, k) = h_1(1, k)$.

- Expansionsregel / Kontrollfunktion

- (a) Geben Sie die nächsten 5 Hashfunktionen obiger Folge an. Wie ist das allgemeine Bildungsgesetz dieser Hashfunktionenfolge (d.h. geben Sie die Formel für $h_L(n, k)$ an)?
- (b) Fügen Sie in die Hashtabelle ($N = 1$, Primärseiten Größe 2, Überlaufseiten Größe 1, Expansionsregel Speicherplatzausnutzung $\alpha = 0.7$) die Schlüssel 73, 40, 50, 11, 63, 51, 71, 75, 4, 55, 39 und 85 ein. Stellen Sie den Zustand der Hashtabelle (Primärseiten, Überlaufseiten) nach jeder Expansion und nach der letzten Einfügung graphisch dar.
(Hinweis: Verwenden Sie die Speicherplatzausnutzung, nicht den Belegungsfaktor – expandieren Sie, falls die Kontrollfunktion verletzt werden würde.)

Zur Wiederholung:

Eine Sortierung von Objekten basiert auf der Definition einer vollständigen Ordnung “<” auf dem Wertebereich der Objekte. Eine vollständige Ordnung ist u.a. dadurch charakterisiert, dass für zwei beliebige Objekte a und b genau eine der folgenden Beziehungen gilt:

- (a) $a < b$
- (b) $b < a$
- (c) $a = b$

Diese Eigenschaft ist auch bekannt als “*Gesetz der Trichotomie*” (obwohl “*Tritomie*” eigentlich der korrektere Name wäre).

Üblicherweise nimmt man auch an, dass für drei beliebige Objekte a , b und c gilt: Wenn $a < b$ und $b < c$, dann $a < c$. Diese Eigenschaft ist bekannt als “*Transitivität*”. Als Ergebnis eines Sortierverfahrens auf n Schlüsseln K_i , $i = 1, \dots, n$ erwartet man eine Permutation $p(1), \dots, p(n)$ der Indizes, so dass die Schlüssel in nicht-absteigender Ordnung angeordnet sind:

$$K_{p(1)} \leq K_{p(2)} \leq \dots \leq K_{p(n)}$$

Ein Sortierverfahren kann zusätzlich *stabil* sein. Dann wird von der Permutation p verlangt, dass gilt:

$$(K_{p(i)} = K_{p(j)} \wedge i < j) \Rightarrow p(i) < p(j)$$

Aufgabe 9-2 *Stabilität von Sortierverfahren*

4 Punkte

Entscheiden Sie, ob folgende Aussagen zutreffen und begründen Sie Ihre Entscheidung:

- (a) “Sortieren durch Abzählen ist stabil.”
- (b) “Sortieren durch direktes Einfügen (Insertion-Sort) ist stabil.”

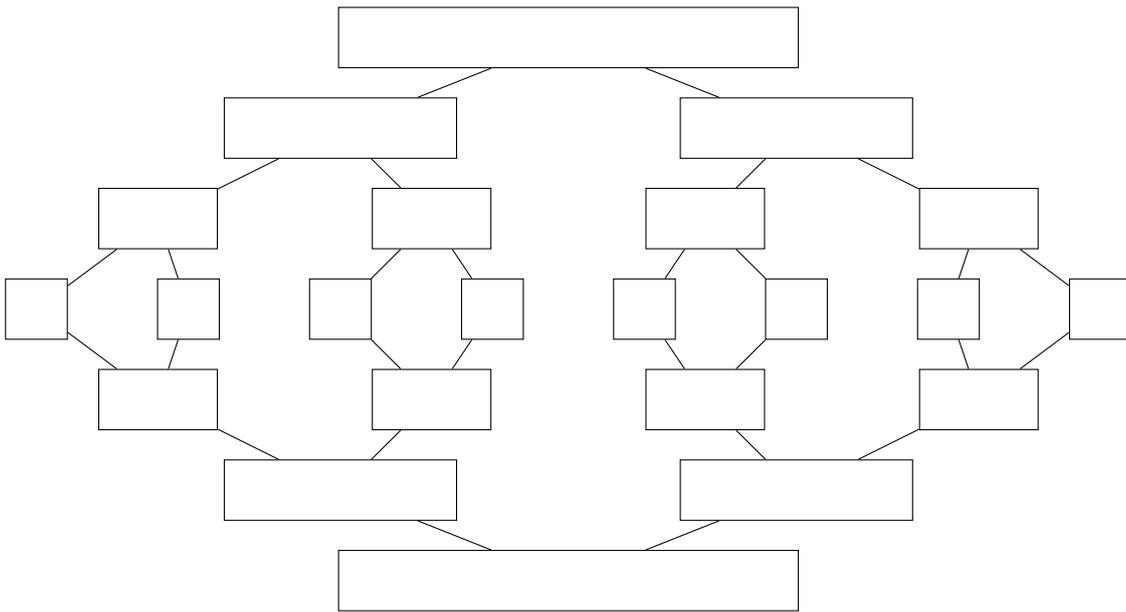
Aufgabe 9-3 *Merge-Sort*

2 Punkte

Sortieren Sie das folgende Array mit dem Merge-Sort-Verfahren *exakt wie es in der Vorlesung besprochen wurde*.

Geben Sie die Zwischenergebnisse des Merge-Sort-Algorithmus in dem untenstehenden Graphen an, der die in der Vorlesung verwendete Struktur besitzt. Die obere Hälfte des Graphen stellt die verschiedenen Rekursionsstufen des Divide-Schritts dar, der obere Knoten des Graphen enthält das unsortierte Array. In die untere Hälfte des Graphen soll der Merge-Schritt eingetragen werden.

Zu sortieren ist die folgende Liste von Zahlen: 14, 84, 75, 25, 49, 45, 64, 78



Aufgabe 9-4 *Quick-Sort*

4 Punkte

Sortieren Sie das folgende Array mit dem Quick-Sort-Verfahren *exakt wie es in der Vorlesung besprochen wurde*: Wählen Sie stets das erste Element als Pivot-Element. Verwenden Sie eine Tabelle mit der folgenden Struktur:

Anfang:	34	71	1	22	26	12	14	30	40	49	53	92
Vor Pivot:												
Nach Pivot:												
Vor Pivot:												
Nach Pivot:												

...

Markieren Sie für jeden rekursiven Durchlauf des Verfahrens das betroffene Intervall *vor und nach* nach dem Verschieben des Pivot-Elements. Es reicht, wenn Sie das Pivot-Element und diejenigen Elemente eintragen, deren Position sich ändert!

Streichen Sie *nie* richtige Elemente durch: Durchgestrichenes können wir nicht bewerten, die Lösung muss für uns eindeutig erkennbar sein! Eigene Markierungen (Zeiger, Vertauschungen etc.) können Sie zwischen den Zeilen eintragen.