**Ludwig-Maximilians-Universität München**
**Department Institut für Informatik**
**Lehr- und Forschungseinheit für Datenbanksysteme**
Prof. Dr. Hans-Peter Kriegel
Dr. Elke Achtert, Tobias Emrich

München, 22.06.2010

# Algorithmen und Datenstrukturen

SS 2010

## Übungsblatt 10: Sortierverfahren

Besprechung: 29.06.2010 - 02.07.2010

**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 $K_i$, $i = 1, \ldots, n$ erwartet man eine Permutation $p(1), \ldots, p(n)$ der Indizes, so dass die Schlüssel in nicht-absteigender Ordnung angeordnet sind:

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

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 10-1** *Stabilität von Sortierverfahren*

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 10-2** *Eindeutigkeit von Sortierverfahren*

(a) Gegeben sei ein *stabiles* Sortierverfahren, basierend auf einer vollständigen Ordnung, die den Gesetzen der *Trichotomie* und *Transitivität* genügt. Beweisen Sie, dass die Permutation $p(1), p(2), \ldots, p(n)$, die man als Ergebnis des Sortierverfahrens erhält, *eindeutig bestimmt* ist.

(b) Nun sei eine Ordnung "$<$" auf $K_1, \ldots, K_n$ gegeben, die dem Gesetz der *Trichotomie* genügt, aber *nicht transitiv* ist. Begründen Sie, warum es auch möglich ist, die Schlüssel *stabil* zu sortieren, wenn die vorausgesetzte Ordnung *nicht transitiv* ist. Nennen Sie zwei Beispiele für Sortierverfahren aus der Vorlesung, die im nicht-transitiven Fall eine stabile Sortierung ermöglichen.

**Aufgabe 10-3** *Quick-Sort*

(a) Geben Sie eine Auswertung von Quick-Sort mit dem Array $a = \{4, 6, 2, 8, 3, 1, 7, 9, 5, 1\}$ an. Die Auswertung soll für alle Aufrufe der Methode *sort* (siehe Skript) die aktuellen Werte der Variablen $li, r, i, j$, sowie die Werte im Array verfolgen. Fertigen Sie dazu eine Tabelle nach folgendem Muster an, in der eine Zeile die jeweils neuen Werte enthält:

| Aufruf *sort* | $li$ | $r$ | $i$ | $j$ | $a1$ | $a2$ | $a3$ | $a4$ | $a5$ | $a6$ | $a7$ | $a8$ | $a9$ | $a10$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 10 | 2 | 10 | 4 | 6 | 2 | 8 | 3 | 1 | 7 | 9 | 5 | 1 |
| | | | | | | 1 | | | | | | | | 6 |
| ... | | | | | | | | | | | | | | |
| 1.1 | 1 | ... | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | |

Ein Vordruck der leeren Tabelle findet sich auf der nächsten Seite. Für das Ausfüllen sind evtl. nicht alle Zeilen notwendig.

(b) Ist Quick-Sort stabil? Begründen Sie Ihre Antwort.

| Aufruf *sort* | $li$ | $r$ | $i$ | $j$ | $a1$ | $a2$ | $a3$ | $a4$ | $a5$ | $a6$ | $a7$ | $a8$ | $a9$ | $a10$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 10 | 2 | 10 | 4 | 6 | 2 | 8 | 3 | 1 | 7 | 9 | 5 | 1 |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |