Fachbereich IV - Abteilung Informatik
Algorithmen und Datenstrukturen
Universität Trier

Prof. S. Näher Tel.: 0651 201-3275

6. Übung zur Vorlesung:

Algorithmen und Datenstrukturen

Sommersemester 2009 10. Juni 2009

Abgabe bis Montag, 22. Juni 2009, 10:00 h, im Briefkasten vor H426

Aufgabe 6.1: (Punkte 5)

Zeigen Sie, dass man n ganze Zahlen aus dem Bereich zwischen 0 und $n^2 - 1$ in Zeit und Platz $O(n)$ sortieren kann. Hinweis: Behandeln Sie die(Binärdarstellungen der) Zahlen wie Strings der Länge 2.

Aufgabe 6.2: (Punkte 5)

(Bottom-Up-Heapsort)
Schreiben Sie einen Algorithmus zum Sortieren durch *Bottom-Up-Heapsort*. Dazu soll die $SINK$ Funktion so geändert werden, dass zunächst der potentielle Sinkpfad bis zu einem Blatt verfolgt wird und dann die richtige Position des Schlüssels durch Aufsteigen im Heap ermittelt wird. Wie viele Vergleiche sind notwendig ?

Aufgabe 6.3: (Punkte 5)

(Externes Sortieren)
Die in der Vorlesung behandelten Sortierverfahren gehen davon aus, dass alle Daten in den Hauptspeicher des Computers passen (als Array). In der Praxis ist dies oft nicht der Fall. Dann verwendet man sogenannte *externe* Sortierverfahren. Nehmen Sie an, der Hauptspeicher hat Größe m. Überlegen Sie sich einen Sortieralgorithmus, der mithilfe des Hauptspeichers sortierte Blöcke der Größe m erzeugt und diese dann mithilfe von Dateien auf der Platte zusammen mischt.

Aufgabe 6.4: (Punkte 5)

(Paralleles Sortieren)
Gegeben seien n (n sei gerade) Prozessoren $p_1, \ldots, p_n$. Prozessor p_i ($2 \leq i \leq n-1$) habe die Möglichkeit mit den Prozessoren p_{i-1} und p_{i+1} zu kommunizieren. p_1 kommuniziert nur mit p_1 und p_n nur mit p_{n-1}. Jeder Prozessor speichert genau einen Wert. In jedem Rechenschritt kann jeder Prozessor nur mit genau einem Prozessor kommunizieren. In einem solchen Schritt werden die beiden Werte verglichen. Der kleinere der beiden Werte wird dem Prozessor mit dem kleineren Index zugeordnet. Der grössere Wert wird dem Prozessor mit dem grösseren Index zugeordnet.
Betrachten Sie nun das Verfahren (**odd-even transposition sort**), das aus wiederholter Ausführung der beiden Schritte (a) und (b) besteht.

a) p_{2i-1} kommuniziert mit p_{2i} für alle $i \in \{1, \ldots, \frac{n}{2}\}$ gleichzeitig.

b) p_{2i} kommuniziert mit p_{2i+1} für alle $i \in \{1, \ldots, \frac{n}{2} - 1\}$ gleichzeitig.

Überlegen Sie sich, wie oft man die beiden Schritte ausführen muss, um eine beliebige Anfangsfolge der Länge n sortieren zu können, wobei das i-te Element der Anfangsfolge anfangs in Prozessor i steht. Setzen Sie hierbei den Satz voraus, dass der Algorithmus genau dann jede Folge korrekt sortiert, wenn er die Folge $(n, n-1, n-2, \ldots, 3, 2, 1)$ korrekt sortiert.