

6. Übung:

Algorithmen und Datenstrukturen

Sommersemester 2010

7. Juni 2010

Abgabe bis Montag, 14. Juni 2010, 10:00 im Briefkasten vor H426

Aufgabe 6.1: (Punkte 6)

Entwickeln Sie eine *stabile* Variante von Counting-Sort, die im letzten Schritt das Eingabefeld von vorne nach hinten und nicht (wie in der Vorlesung) von hinten nach vorne liest. Hinweis: Überlegen Sie sich, wie die Definition der Werte im Feld C geändert werden muss.

Aufgabe 6.2: (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 6.3: (Punkte 5)

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 6.4: (Punkte 5)

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.