Technische Universität München
Institut für Informatik
Lehrstuhl für Effiziente Algorithmen

WS 2002/2003
Übungsblatt 7
2. Dezember 2002

Grundlegende Algorithmen

Abgabe: **Mittwoch, 4. Dezember**, *vor der Vorlesung,* **MW0350**

Aufgabe 1

Betrachten Sie folgenden Sortieralgorithmus

(0) CountingSort($A[], n, N$)
(1) `begin`
(2)   `for` $i := 0$ `to` $(n-1)$ `do` $C[A[i]] := (C[A[i]] + 1)$;
(3)   `for` $i := 1$ `to` N `do` $C[i] := (C[i] + C[i-1])$;
(4)   `for` $i := (n-1)$ `downto` 0 `do begin`
(5) $B[C[A[i]] - 1] := A[i]$;
(6) $C[A[i]] := (C[A[i]] - 1)$;
(7)   `end`;
(8)   `for` $i := 0$ `to` $(n-1)$ `do` $A[i] := B[i]$
(9) `end`.

CountingSort sortiert das Feld $A[]$ korrekt, wenn N mindestens so groß wie das Maximum aller Feldelemente ist.

(a) Geben Sie für die Eingabe $(0, 3, 7, 5, 4, 3, 3, 2, 8, 9, 1, 0)$ die Inhalte der Felder B und C

- nach Abarbeitung von Zeile (2)
- nach Abarbeitung von Zeile (3)
- nach erster, zweiter und dritter Iteration in Zeilen (4) bis (7)
- vor Abarbeitung von Zeile (8)

an.

(b) Bestimmen Sie in Abhängigkeit von n und N die Anzahl arithmetischer Operationen von CountingSort. Interpretieren und diskutieren Sie das Ergebnis im Vergleich mit Satz 23 aus der Vorlesung (untere Schranke für das Sortieren).

(c) Ersetzen Sie Zeile (4) durch:

(4') `for` $i := 0$ `to` $(n-1)$ `do begin`

Zeigen Sie, dass der Algorithmus immer noch korrekt ist.

Aufgabe 2

Zeigen Sie: Genügen zur Bestimmung des Medians einer Menge mit n Elementen maximal $c \cdot n$ Vergleiche, so benötigt die Variante von QuickSort, die als Pivot den Median auswählt, maximal $(c+1)n \log n + O(n)$ Vergleiche.