

Algorithmen und Datenstrukturen

Wintersemester 2005/06

2. Übungsblatt

Aufgabe 5:

Der MERGE-SORT Algorithmus soll auf das Eingabefeld $\langle 3, 41, 52, 26, 38, 57, 9, 49 \rangle$ angewandt werden. Skizzieren Sie den Rekursionsbaum der MERGE-Phase.

Aufgabe 6:

Beweisen Sie durch vollständige Induktion: Ist n eine Zweierpotenz, so ist die Funktion $T(n) = n \log_2 n$ die eindeutige Lösung der Rekursionsgleichung

$$T(n) = \begin{cases} 2 & \text{falls } n = 2, \\ 2T(n/2) + n & \text{falls } n = 2^p, p > 1. \end{cases}$$

Aufgabe 7:

Man kann INSERTION-SORT (auch wenn dies etwas künstlich anmutet) als rekursiven Algorithmus ausdrücken: um $A[1..n]$ zu sortieren sortieren wir zuerst $A[1..n-1]$ und fügen daraufhin $A[n]$ in das sortierte Feld $A[1..n-1]$ ein. Stellen Sie eine Rekursionsgleichung für die Laufzeit dieser rekursiven Umformulierung von INSERTION-SORT auf.

Aufgabe 8:

Sei $A[1..n]$ ein Feld aus n paarweise verschiedenen ganzen Zahlen, das aufsteigend sortiert werden soll. Um den Aufwand eines Sortierverfahrens abzuschätzen, benutzt man verschiedene *Vorsortierungsmaße*:

- (i) $\text{inv}(A) := |\{(i, j) : 1 \leq i < j \leq n, A[i] > A[j]\}|$ zählt die *Inversionen* (Fehlstellungen) in A .
- (ii) $\text{runs}(A) := |\{i : 1 \leq i < n, A[i+1] < A[i]\}| + 1$ zählt die bereits aufsteigend sortierten Teilfolgen in A .
- (iii) $\text{rem}(A) := n - \text{las}(A)$, wobei $\text{las}(A)$ die Länge der längsten aufsteigend sortierten Teilfolge von A bezeichnet.

(a) Berechnen Sie $\text{inv}(A)$, $\text{runs}(A)$ und $\text{rem}(A)$ für das Feld

$$A = \langle 2, 5, 6, 1, 4, 3, 8, 9 \rangle.$$

(b) Welches Zahlenfeld, das die Zahlen $1, \dots, n$ enthält, besitzt die größtmögliche Zahl von Inversionen? Wieviele Inversionen besitzt dieses Feld?

(c) Welche Beziehung besteht zwischen der Laufzeit von INSERTION-SORT und der Anzahl Inversionen des Eingabefeldes? Begründen Sie Ihre Antwort.

(d) Berechnen Sie $\text{inv}(A_i)$, $\text{runs}(A_i)$ und $\text{rem}(A_i)$ für die Felder A_i , $i = 1, 2, 3$, jeweils in Abhängigkeit von n , n gerade.

$$A_1 = \langle 2, 1, 4, 3, 6, 5, \dots, n, n-1 \rangle$$

$$A_2 = \langle n/2 + 1, n/2 + 2, n/2 + 3, \dots, n, 1, 2, 3, \dots, n/2 \rangle$$

$$A_3 = \langle 1, n, 2, n-1, 3, n-2, \dots, n/2, n/2 + 1 \rangle$$

Diskutieren Sie Vor- und Nachteile der drei Vorsortierungsmaße.

Aufgabe 9: (Programmieraufgabe)

Die Rekursion im MERGE-SORT-Algorithmus läßt sich beseitigen. Betrachten Sie dazu folgende iterative Variante von MERGE-SORT für den Spezialfall $n = 2^p$, $p \geq 0$:

I-MERGE-SORT(A)

```
1   $\ell \leftarrow 1$ 
2   $k \leftarrow n \leftarrow \text{length}[A]$ 
3  for  $i \leftarrow 1$  to  $\lg(n)$ 
4      do  $\ell \leftarrow 2 \cdot \ell$ 
5           $k \leftarrow k/2$ 
6          for  $j \leftarrow 1$  to  $k$ 
7              do  $p \leftarrow (j-1) \cdot \ell + 1$ 
8                   $r \leftarrow p + \ell - 1$ 
9                   $q \leftarrow \lfloor (p+r)/2 \rfloor$ 
10                 MERGE( $A, p, q, r$ )
```

Implementieren Sie MERGE-SORT und I-MERGE-SORT in der Programmiersprache C, wobei I-MERGE-SORT nur für den Fall $n = 2^p$, $p \geq 0$, arbeiten soll.

Realisieren Sie beide Sortieralgorithmen jeweils als Unterprogramm mit geeigneten Parametern. Das zu sortierende Eingabefeld A bestehe aus ganzen Zahlen (Integer) und ist aus einer Textdatei zu lesen. Im Hauptprogramm soll das Feld A mit beiden Algorithmen sortiert werden.