INSTITUT FÜR INFORMATIK
KOMPLEXITÄT UND KRYPTOGRAFIE
PROF. DR. JOHANNES KÖBLER

SS 2008
14. MAI 2008

# Theoretische Informatik 3

***3. Übung***

**Abgabe der schriftlichen Lösungen bis zum 4. Juni 2008**

**Aufgabe 13** [*5 Punkte*]

Lösen Sie die folgenden Rekursionsgleichungen, indem Sie möglichst enge untere und obere asymptotische Schranken für die Lösungsfunktionen angeben (evtl. in Abhängigkeit vom Anfangswert $T(1) = a$).

a) $T(n) = T(n/2) + T(n/3) + T(n/4) + dn^k$, $d, k \geq 0$, (*mündlich*)

b) $T(n) = bT(n-c) + dn + e$, $b \geq 0$, $c \geq 1$, (*mündlich*)

c) $T(n) = bT(cn) + dn + e$, $b \geq 0$, $0 < c < 1$. (*5 Punkte*)

**Aufgabe 14** [*mündlich*]

Implementieren Sie `MergeSort` als nichtrekursives "in place"-Sortierverfahren.

*Hinweis:* Die zu sortierende Zahlenfolge soll durch einen Zeiger auf eine einfach verkettete Liste $L$ übergeben werden. Bei jedem Schleifendurchlauf soll der Algorithmus die Liste $L$ zuerst so in zwei Listen $A$ und $B$ zerlegen, dass sortierte Teilstücke maximaler Länge (so genannte *Runs*) zusammenbleiben und abwechselnd auf die beiden Listen $A$ und $B$ verteilt werden. Im zweiten Teil der Schleife sollen die Listen $A$ und $B$ wieder zu einer Liste gemischt werden, so dass sich die Anzahl der Runs bei jedem Schleifendurchlauf halbiert.

**Aufgabe 15** [*mündlich*]

Die Prozedur `Merge` benötigt im schlechtesten Fall $n+m-1$ Vergleiche, um zwei sortierte Zahlenfolgen $a_1 \leq \cdots \leq a_n$ und $b_1 \leq \cdots \leq b_m$ zu einer sortierten Folge $c_1 \leq \cdots \leq c_{n+m}$ zusammenzufügen.

a) Zeigen Sie, dass es keine Prozedur gibt, die dies im Fall $n = m$ mit weniger als $2n-1$ Vergleichen schafft.

b) Überlegen Sie sich ein Verfahren, das im Fall $m \ll n$ (z.B. $m \approx \sqrt{n}$) mit deutlich weniger Vergleichen auskommt.

**Aufgabe 16** [*mündlich*]

Die *Tiefe* $t(v)$ eines Knotens in einem Baum mit Wurzel $w$ sei die Länge des Pfades von $w$ zu $v$. Sei $B$ ein Binärbaum mit $n$ Blättern. Zeigen Sie:

a) Die maximale Blatttiefe von $B$ beträgt mindestens $\lceil \log_2 n \rceil$.

b) Die mittlere Blatttiefe von $B$ beträgt mindestens $\log_2 n$.

c) Ein vergleichendes Sortierverfahren benötig im Durchschnitt mindestens $\log_2(n!)$ Vergleiche, um eine zufällige Permutation der Zahlen $1, \ldots, n$ zu sortieren.

**Aufgabe 17** [*mündlich*]

a) Geben Sie den Rekursionsbaum von `QuickSort` für die Folge $(3, 1, 5, 2, 4)$ an.

b) Wie viele Fragen benötigt `QuickSort`, um die Folge $(3, 1, 5, 2, 4)$ zu sortieren?

c) Geben Sie den Fragebaum von `QuickSort` für die Sortierung von $4$ Zahlen an.

d) Bestimmen Sie für $n = 1, \ldots, 10$ die Fragekomplexität von `QuickSort` im besten, schlechtesten und durchschnittlichen Fall für eine Permutation der Zahlen $1, \ldots, n$. Vergleichen Sie diese Werte jeweils mit der unteren Schranke $\lceil \log_2(n!) \rceil$ für den schlechtesten Fall.

e) Bestimmen Sie für alle Paare $1 \leq i < j \leq 5$ das erste von `QuickSort` beim Sortieren der Folge $(3, 1, 5, 2, 4)$ im Intervall $I_{ij} = \{i, \ldots, j\}$ gewählte Pivotelement.

**Aufgabe 18** [*mündlich*]

Zeigen Sie, dass `QuickSort` im Durchschnitt $V(n) = 2n \ln n - \Theta(n) = (2 \ln 2) \log_2(n!) - \Theta(n)$ Vergleiche benötigt, um eine Folge von $n$ paarweise verschiedenen Zahlen zu sortieren.

*Hinweis:* Für die *harmonische Reihe* $H_n = \sum_{i=1}^{n} \frac{1}{i}$ gilt $\ln(n+1) \leq H_n \leq \ln n + 1$.

**Aufgabe 19** [*mündlich*]

Der Algorithmus `RandomQuickSort` ist eine randomisierte Variante von `QuickSort`, bei der nicht das letzte Element des zu sortierenden Arrays sondern ein zufälliges Element als Pivot-Element gewählt wird.

a) Bestimmen Sie die (erwartete) Anzahl von Vergleichen, die `RandomQuickSort` im besten, mittleren und schlechtesten Fall vornimmt, um eine Folge von $n$ paarweise verschiedenen Zahlen zu sortieren.

*Hinweis:* Zeigen Sie, dass sich `RandomQuickSort` bei jeder Eingabefolge verhält wie `QuickSort` bei einer zufälligen Permutation dieser Folge.

b) Welche Erwartungswerte ergeben sich im besten und im schlechtesten Fall, wenn die Folgenglieder mehrfach vorkommen können?

**Aufgabe 20** [*2+3 Punkte*]

a) Wie viele Fragen benötigt `HeapSort`, um die Folge $(3, 1, 5, 2, 4)$ zu sortieren?

b) Geben Sie den Fragebaum von `HeapSort` für die Sortierung von $4$ Zahlen an.