Johannes Blömer
u.v.a.m.

Paderborn, den 20. Mai
Abgabe 2. Juni 2004, 11:00, Kästen Flur D3

Übungen zur Vorlesung
Datenstrukturen und Algorithmen
SS 2004
Blatt 5

AUFGABE 1 (4 Punkte)**:**

1. Was ist in $\mathcal{O}$-Notation die Laufzeit von Quicksort, wenn n identische Zahlen sortiert werden sollen? (2 Punkte)

2. Was ist in $\mathcal{O}$-Notation die Laufzeit von Quicksort, wenn die Eingabezahlen absteigend sortiert sind? (2 Punkte)

AUFGABE 2 (4 Punkte)**:**
Angenommen, die Funktion PARTITION teilt Teilarrays der Größe n immer in zwei Teilarrays auf, von denen eines $(1-\alpha)n$ Elemente und das andere αn Elemente enthält, hierbei ist $\frac{1}{2} \leq \alpha < 1$. Zeigen Sie, dass dann jedes Blatt im Rekursionsbaum für Quicksort höchstens Tiefe $-\log(n)/\log(\alpha)$ und mindestens Tiefe $-\log(n)/\log(1-\alpha)$ besitzt. In der gesamten Aufgabe dürfen Auf- bzw. Abrundungen ignoriert werden.

AUFGABE 3 (12 Punkte)**:**
Betrachten Sie den folgenden alternativen Algorithmus zur Partitionierung eines Teilarrays, wie er von Quicksort benutzt wird.

HOARE-PARTITION(A, p, r)

```
x ← A[p]
i ← p − 1
j ← r + 1
while i < j
        do  repeat j ← j − 1
              until A[j] ≤ x
            repeat i ← i + 1
              until A[i] ≥ x
            if i < j
              then A[i] ↔ A[j]
return j
```

Zeigen Sie

1. Der Algorithmus greift nie auf Elemente $A[i], A[j]$ ausserhalb des Teilarrays $A[p..r]$ zu. (4 Punkte)

2. Für den Wert j, den Algorithmus HOARE-PARTITION ausgibt, gilt $p \leq j \leq r - 1$. (4 Punkte)

3. Am Ende des Algorithmus HOARE-PARTITION und für den Ausgabewert j gilt: *Alle Elemente in $A[p..j]$ sind höchstens so gross wie die Elemente in $A[j+1..r]$.* (4 Punkte)