

Datenstrukturen und Algorithmen

Übung 2

3. März 2011

Diese Übung muss zu Beginn der Übungsstunde am 10. März abgegeben werden. Die Abgabe der DA Übungen erfolgt immer in schriftlicher Form auf Papier. Programme müssen zusammen mit der von ihnen erzeugten Ausgabe abgegeben werden. Drucken Sie wenn möglich platzsparend 2 Seiten auf eine A4-Seite aus. Falls Sie mehrere Blätter abgeben heften Sie diese bitte zusammen (Büroklammer, Bostitch, Mäppchen). Alle Teilnehmenden dieser Vorlesung müssen Ihre eigene Übung abgeben (einzeln oder in Zweiergruppen). Vergessen Sie nicht, Ihren Namen und Ihre Matrikelnummer auf Ihrer Abgabe zu vermerken.

Theoretische Aufgaben

1. Erklären Sie, warum die Aussage “Die Laufzeit von Algorithmus A ist mindestens $O(n^3)$ ”, keinen Sinn macht. **(1 Punkt)**
2. Zeigen Sie, dass die Laufzeit eines Algorithmus genau dann in $\Theta(g(n))$ ist, wenn seine Worst Case Laufzeit in $O(g(n))$ und seine Best Case Laufzeit in $\Omega(g(n))$ ist. **(1 Punkt)**
3. Verwenden Sie die Rechenregeln für Logarithmen, um zu zeigen dass $a^{\log_b n} = n^{\log_b a}$. **(1 Punkt)**
4. Verwenden Sie die Rechenregeln für Logarithmen, um zu zeigen dass $\Theta(\log_a n) = \Theta(\log_b n)$. **(1 Punkt)**
5. Zeigen Sie mittels Induktion, dass die Rekursionsgleichung $T(n) = 2T(\lceil n/4 \rceil + 12) + 3n$ die Lösung $O(n \log n)$ hat. **(1 Punkt)**
6. Zeichnen Sie einen Rekursionsbaum für die Gleichung $T(n) = T(n/3) + T(2n/3) + cn$. Erklären Sie anhand des Baumes, dass die Lösung der Gleichung in $\Omega(n \log n)$ ist. **(1 Punkt)**
7. Die Rekursionsgleichung für die Zeitkomplexität der binären Suche ist $T(n) = T(n/2) + \Theta(1)$. Verwenden Sie die Mastermethode, um zu zeigen dass $T(n) = \Theta(\log n)$. **(1 Punkt)**
8. Berechnen Sie die lösbare Problemgrösse in der gegebenen Zeit für Algorithmen mit verschiedener Zeitkomplexität, welche in der Tabelle gegeben sind. Nehmen Sie an, jede Operation dauere 0.01s. **(2 Punkte)**

Zeitkomplexität $T(n)$	Problemgrösse lösbar in 10s	Problemgrösse lösbar in 1000s
$5n$		
$3n^5$		
$2n^{1.1}$		
$5 \log_2(7n)$		
2^{4n}		

9. Geben Sie die asymptotische Laufzeit dieses Algorithmus in Abhängigkeit von n an. Verwenden Sie die Theta-Notation. Geben Sie eine Summenformel für die Laufzeit an. Hinweis: Verwenden Sie die Summenformel $\sum_{i=0}^k q^i = \frac{q^{k+1}-1}{q-1}$. (1 Punkt)

```

1   $i \leftarrow 1$ 
2  while  $i < n$ 
3      do
4           $j \leftarrow 0$ 
5          while  $j \leq i$ 
6              do
7                   $j \leftarrow j + 1$ 
8           $i \leftarrow 2i$ 

```