Datenstrukturen und Algorithmen
Übung 2

1. März 2018

Abgabe: Diese Übung muss zu Beginn der Übungsstunde bis spätestens um 16 Uhr 15 am 8. 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). *Der gesammte Sourcecode muss ausserdem elektronisch über Ilias abgegeben werden.*

Die Übung sollte vorzugsweise in Zweiergruppen bearbeitet werden, kann aber auch einzeln abgegeben werden. Jede Übungsserie gibt 10 Punkte. Im Durchschnitt müssen Sie 7 von 10 Punkten erreichen, um die Testatbedingung zu erfüllen.

1. Grundlegende Definitionen:

   (a) Erklären Sie, warum die Aussage "Die Laufzeit von Algorithmus A ist mindestens $O(n)$", keinen Sinn macht. (**1 Punkt**)

   (b) 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**)

2. Verwenden Sie die Rechenregeln für Logarithmen...

   (a) ...um zu zeigen dass $a^{\log_b n} = n^{\log_b a}$.

   (b) ...um zu zeigen dass $\Theta(\log_a n) = \Theta(\log_b n)$.

   (c) Gilt auch $\Theta(a^n) = \Theta(b^n)$ wenn $0 < a < b$? Begründen Sie.

   (**2 Punkte**)

3. Zeigen Sie mittels Induktion/der Substitutionsmethode, dass die Rekursionsgleichung $T(n) = 2T(\lceil n/4 \rceil + 12) + 3n$ die Lösung $O(n)$ hat. (**1 Punkt**)

4. 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**)

5. 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**)

6. 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 $1000s$
$10n$		
$2n^3$		
$n^{2.5}$		
$2\log_2(8n)$		
2^{2n}		

7. 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 Partialsummenformel für geometrische Reihen ($\sum_{k=0}^{n} a_0 q^k = a_0 \frac{q^{n+1}-1}{q-1}$). (**1 Punkt**)

```
i ← 1
while i < n
    do
        j ← 0
        while j ≤ i
            do
                j ← j + 1
        i ← 3i
```