Prof. Dr. Leif Kobbelt
Thomas Ströder, Fabian Emmes
Sven Middelberg, Michael Kremer

Informatik 8
RWTH Aachen

Datenstrukturen und Algorithmen (SS 2013)

Übungsblatt 4

Abgabe: Montag, **13.05.2013**, 14:00 Uhr

- Die Übungen sollen in Gruppen von zwei bis drei Personen bearbeitet werden.
- Schreiben Sie die Namen jedes Gruppenmitglieds sowie alle Matrikelnummern auf die abgegebenen Lösungen.
- Schreiben Sie die Namen jedes Gruppenmitglieds sowie alle Matrikelnummern auch in die Quellcode-Dateien.
- Geben Sie Ihre Lösungen am **Anfang** der Globalübung, montags, 14:00 Uhr, ab.
- Schicken Sie den jeweiligen Quellcode bitte per **E-Mail** direkt an Ihre/n Tutor/in.
- Geben Sie außerdem den ausgedruckten Quellcode zusammen mit den schriftlichen Lösungen ab.
- Zu spät abgegebene Lösungen werden nicht bewertet.
- Sofern nicht anders gefordert, müssen alle Lösungen und Zwischenschritte kommentiert werden.

Aufgabe 1 (Algorithmische Analyse [**10 Punkte**])

1. Beweisen Sie die folgenden Eigenschaften der Komplexitätsklassen O, Ω und Θ: Für alle $f, g, h : \mathbb{N} \to \mathbb{R}^+$ gilt [3 Punkte]

   - **Transitivität:** $f \in O(g) \wedge g \in O(h) \Rightarrow f \in O(h)$, analog für Ω und Θ
   - **Reflexivität:** $f \in O(f)$, $f \in \Omega(f)$, $f \in \Theta(f)$
   - **Symmetrie:** $f \in \Theta(g) \Rightarrow g \in \Theta(f)$
   - **Antisymmetrie:** $f \in O(g) \wedge g \in O(f) \Rightarrow f \in \Theta(g)$ und analog
     $f \in \Omega(g) \wedge g \in \Omega(f) \Rightarrow f \in \Theta(g)$

2. Beschreiben Sie das asymptotische Verhalten folgender Funktionen möglichst genau:

   - $T(n) = 2T(n/2) + n^3$
   - $T(n) = 16T(n/4) + n^2$
   - $T(n) = 3T(n/4) + n \log n$
   - $T(3n) = 3T(2n) + \log n$

   Hierbei sei stets $T(1) = 1$. [3 Punkte]

3. Zeigen oder widerlegen Sie [1 Punkt]:

$$f \in \Theta(g) \wedge \lim_{n\to\infty} \frac{h(n)}{g(n)} = 0 \quad \Rightarrow \quad f + h \in \Theta(g).$$

4. Zeigen oder widerlegen Sie [1 Punkt]:

$$f \in O(n \log_a n) \quad \Leftrightarrow \quad f \in O(n \log_b n).$$

5. Lösen Sie die folgende Rekursionsgleichung so auf, dass sie keine Rekursion mehr enthält, d.h. nur noch von n, nicht aber von $T(n)$ abhängig ist. Die Ermittlung der Komplexitätsklasse ist nicht ausreichend. [2 Punkte]:

$$\begin{aligned} T(0) &= 5 \\ 2T(n) &= nT(n-1) + 3 \cdot n! \end{aligned}$$

Aufgabe 2 (Rekursionsgleichungen [**10 Punkte**])

Gegeben sei folgende Java-Methode (der Einfachheit halber kann angenommen werden, dass der Parameter `i` stets positiv oder null ist):

```
public static int func(int i) {

  if(i == 0 || i == 1) return i;

  int f1 = func((i+1)/2);
  int f2 = func((i+1)/2 - 1);

  if(i % 2 == 1) {
    return f1 * f1 + f2 * f2;
  }

  return f1 * f1 + 2 * f1 * f2;
}
```

Beachten Sie, dass bei der Ganzzahldivision immer auf die nächstniedrigere Ganzzahl *abgerundet* wird.

(a) Bestimmen Sie die Rekursionsgleichung für die Funktion. [4 Punkte]

(b) Schätzen Sie die asymptotische Komplexität mit Hilfe des Master-Theorems ab. [4 Punkte]

(c) Beschreiben Sie, was die Funktion berechnet. [2 Punkte]