Prof. Dr. E. Ábrahám

F. Corzilius, S. Schupp, T. Ströder

Allgemeine Hinweise:

- Die **Hausaufgaben** sollen in Gruppen von je **2 bis 3 Studierenden** aus der **gleichen Kleingruppenübung (Tutorium)** bearbeitet werden. **Namen und Matrikelnummern** der Studierenden sind auf jedes Blatt der Abgabe zu schreiben. **Heften bzw. tackern Sie die Blätter!**
- Die **Nummer der Übungsgruppe** muss **links oben** auf das **erste Blatt** der Abgabe geschrieben werden. Notieren Sie die Gruppennummer gut sichtbar, damit wir besser sortieren können.
- Die Lösungen müssen **bis Montag, den 05.05.2014 um 9:00 Uhr** in den entsprechenden Übungskasten eingeworfen werden. Sie finden die Kästen am Eingang Halifaxstr. des Informatikzentrums (Ahornstr. 55). Alternativ können Sie die Lösungen auch in Ihrem Tutorium vor der Abgabefrist direkt bei Ihrer Tutorin/Ihrem Tutor abgeben.
- Sofern nicht näher spezifiziert, steht die Notation log ohne Angabe einer Basis für den natürlichen Logarithmus.

Tutoraufgabe 1 (Rekursionsgleichungen):

- Geben Sie die Rekursionsgleichungen für die Laufzeit der folgenden Algorithmen an. Dabei sind die elementaren Operationen $\{+, -, \cdot, \div, \sqrt{\ }, \nearrow^2\} \in \mathcal{O}(1)$. Bei Algorithmen mit Aufruf einer Unterfunktion m lassen Sie die Unterfunktionslaufzeit als Parameter $T_m(\cdot)$ in die eigentliche Gleichung einfließen und geben Sie die Rekursionsgleichungen für m ebenfalls an.
- Geben Sie bei Algorithmen mit mehreren Parametern an, welche der Parameter einen Einfluss auf die Laufzeit haben.

a)

```
int mult(int a, int b)
{
  if(a >= 1)
    return mult(a-1,b) + b;
  else if(a <= -1)
    return mult(a+1,b) - b;
  else // if a==0
    return 0;
}

int collatz(unsigned n)
{
  if(n<=1)
  {
    return 1;
  }
  if(n.isOdd()) // gibt true zurück, wenn n ungerade
  {
    return collatz(mult(3,n) + 1)
  }
  else
  {
    return collatz(n/2);
  }
}
```

http://de.wikipedia.org/wiki/Collatz-Problem

b)
```
float heron(float number, int depth)
{
  if(depth > 0)
  {
    it = depth - 1;
    return (heron(number, it) + number/heron(number, it))/2;
  }
  else
  {
    return (number+1)/2; // Startwert
  }
}
```
http://de.wikipedia.org/wiki/Heron-Verfahren

Aufgabe 2 (Rekursionsgleichungen): (2+2*+2 = 4+2* Punkte)

- Geben Sie die Rekursionsgleichungen für die Laufzeit der folgenden Algorithmen an. Dabei sind die elementaren Operationen $\{+, -, \cdot, \div, \sqrt{}, \nearrow^2\} \in \mathcal{O}(1)$. Bei Algorithmen mit Aufruf einer Unterfunktion m lassen Sie die Unterfunktionslaufzeit als Parameter $T_m(\cdot)$ in die eigentliche Gleichung einfließen und geben Sie die Rekursionsgleichungen für m ebenfalls an.
- Geben Sie bei Algorithmen mit mehreren Parametern an, welche der Parameter einen Einfluss auf die Laufzeit haben.

a)
```
float sierpinski(int depth, float length)
{
  if(depth>0)
  {
    return sierpinski(depth-1, length/2) +
           sierpinski(depth-1, length/2) +
           sierpinski(depth-1, length/2);
  }
  else
  {
    return √(length² − (length/2)²) * (length/2);
    // the area of a triangle with edge-lenght l
  }
}
```
http://de.wikipedia.org/wiki/Sierpinski-Dreieck

b)* Finden Sie eine nicht-rekursive Beschreibung für die Laufzeit des obigen `sierpinski` Algorithmus.

c)
```
int fak(int n)
{
  if (n > 1)
  {
    return n*fak(n-1);
  }
  else
  {
    return 1;
  }
}
```

```
float euler(int depth)
{
  if (depth >= 1)
  {
    return 1/fak(depth) + euler(depth-1);
  }
  else
  {
    return 1;
  }
}
```

http://de.wikipedia.org/wiki/Eulersche_Zahl

Tutoraufgabe 3 (Substitutionsmethode):

Gegeben sei die folgende Rekursionsgleichung:

$$T(n) \quad = \quad \begin{cases} 4 \cdot T(\frac{n}{2}) + \frac{n^2}{\log_2(n)}, & \text{falls } n > 1 \\ 1, & \text{sonst} \end{cases}$$

- Schätzen Sie mit Hilfe des Rekursionsbaumes die Komplexitätsklasse der Laufzeit $T(n)$, d.h., geben Sie eine nicht-rekursive Funktion $f(n)$ mit $T(n) \in \Theta(f(n))$ an.
- Beweisen Sie mit Hilfe der Substitutionsmethode, dass $T(n) \in \mathcal{O}(f(n))$.

Hinweis: Die n–te Partialsumme der harmonischen Reihe ist definiert durch $H_n := \sum_{i=1}^{n} \frac{1}{i}$ und hat folgende Eigenschaften:

i) $\forall n \geq 1.\ \log(n) < H_n$

ii) $\forall n > 1.\ H_n - \log(n) \ < \ H_{n-1} - \log(n-1)$

Aufgabe 4 (Substitutionsmethode): (8 Punkte)

Geben Sie zur Rekursionsgleichung

$$T(n) \quad = \quad \begin{cases} 4 \cdot T(\frac{n}{2}) + n^2 \cdot \log_2(n), & \text{falls } n > 1 \\ 1, & \text{sonst} \end{cases}$$

die Komplexitätsklasse (Θ) an und beweisen Sie dies mit Hilfe der Substitutionsmethode.

Tutoraufgabe 5 (Mastertheorem):

Für alle Rekursionsgleichungen in dieser Aufgabe gelte $T(n) = 1$ falls $n \leq 1$. Bestimmen Sie die Komplexitätsklassen (Θ) der folgenden Rekursionsgleichungen in geschlossener Form mithilfe des Mastertheorems oder begründen Sie, warum das Mastertheorem nicht anwendbar ist.

a) $T(n) = 27 \cdot T(\frac{n}{3}) + n^2 \cdot \log(n)$ falls $n > 1$

b) $T(n) = 16 \cdot T(\frac{n}{4}) + \frac{n^3}{\sqrt{n}}$ falls $n > 1$

c) $T(n) = 2 \cdot T(\frac{n}{2}) + \frac{n}{\log(n)}$ falls $n > 1$

Aufgabe 6 (Mastertheorem): (2 + 3 + 3 = 8 Punkte)

Für alle Rekursionsgleichungen in dieser Aufgabe gelte $T(n) = 1$ falls $n \leq 1$. Bestimmen Sie die Komplexitätsklassen (Θ) der folgenden Rekursionsgleichungen in geschlossener Form mithilfe des Mastertheorems oder begründen Sie, warum das Mastertheorem nicht anwendbar ist.

a) $T(n) = 2 \cdot T(\frac{n}{4}) + \sqrt{n}$ falls $n > 1$

b) $T(n) = 25 \cdot T(\frac{n}{5}) + (n \cdot \log(n))^2$ falls $n > 1$

c) $T(n) = 3 \cdot T(\frac{n}{9}) + \frac{n}{\log(n)}$ falls $n > 1$