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

Lösung:

a) Die Rekursionsgleichung für die Laufzeit T_{mult} von `mult` kann wie folgt aufgestellt werden:

$$T_{mult}(a,b) = \begin{cases} T_{mult}(a-1,b)+2, & \text{wenn } a \geq 1 \\ T_{mult}(a+1,b)+2, & \text{wenn } a \leq -1 \\ 0, & \text{sonst} \end{cases}$$

Aus der obigen Gleichung wird ersichtlich, dass nur der Absolutwert vom ersten Parameter eine Wirkung auf die Laufzeit hat. Daher kann man alternativ auch angeben:

$$T_{mult}(a,b) = \begin{cases} T_{mult}(|a|-1)+2, & \text{wenn } |a| \geq 1 \\ 0, & \text{sonst} \end{cases}$$

Die Laufzeit T_c von `collatz` kann nun durch die folgende Rekursionsgleichung beschrieben werden:

$$T_c(n) = \begin{cases} (n \bmod 2) \cdot (T_c(mult(3,n)+1) + T_{mult}(3,n)+1) + ((n+1) \bmod 2) \cdot T_c(n/2)+1, & \text{wenn } n > 1 \\ 0, & \text{sonst } (n \leq 1) \end{cases}$$

Man kann mittels vollständiger Induktion zeigen, dass `mult(a,b)` das Produkt der Parameter als Rückgabewert zurückgibt. Damit bekommen wir:

$$T_c(n) = \begin{cases} (n \bmod 2) \cdot (T_c(3 \cdot n+1) + T_{mult}(3,n)+1) + ((n+1) \bmod 2) \cdot T_c(n/2)+1, & \text{wenn } n > 1 \\ 0, & \text{sonst (wenn } n \leq 1) \end{cases}$$

Alternativ kann man für die *Überapproximation* der Laufzeit auch das Maximum der beiden Zweige nehmen:

$$T_c(n) = \begin{cases} \max(T_c(3 \cdot n+1) + T_{mult}(3,n)+1,\ T_c(n/2)+1), & \text{wenn } n > 1 \\ 0, & \text{sonst (wenn } n \leq 1) \end{cases}$$

b) Die Laufzeit T_h von `heron` kann durch die folgende Rekursionsgleichung beschrieben werden:

$$T_h(n,d) = \begin{cases} 2 \cdot T_h(n,d-1)+4, & \text{wenn } d > 0 \\ 2, & \text{sonst} \end{cases}$$

Bei dieser Aufgabe hat lediglich der zweite Parameter `depth` einen Einfluss auf die Laufzeit.

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

Lösung:

a) Die Laufzeit T_s von `sierpinski` kann durch die folgende Rekursionsgleichung beschrieben werden:
$$T_s(d,l) = \begin{cases} 3 \cdot T_s(d-1, \frac{l}{2}) + 6 + 2, & \text{wenn } d > 0 \\ 7, & \text{sonst} \end{cases}$$
Bei dieser Aufgabe hat nur der Parameter depth einen Einfluss auf die Laufzeit.

b)* Der Rekursionsbaum hat eine Tiefe von d und somit 3^d viele Blätter, welche alle mit der konstanten Laufzeit 7 behaftet sind. Es gilt nun für d Ebenen (Ebene 0 bis Ebene $d-1$) die Laufzeit 8 für die Anzahl der Knoten auf der jeweiligen Ebene (3^i) aufzusummieren um die Gesamtlaufzeit zu erhalten:

$$T_s(d,l) = \underbrace{3^d}_{\text{Anzahl Blätter}} \cdot 7 + \underbrace{\sum_{i=0}^{d-1} 3^i}_{\text{Anzahl innerer Knoten}} \cdot 8$$

c) Die Laufzeit T_{fak} von `fak` und die Laufzeit T_{euler} von `euler` können durch die folgenden Rekursionsgleichungen beschrieben werden:

$$T_{fak}(n) = \begin{cases} T_{fak}(n-1) + 2, & \text{wenn } n > 1 \\ 0, & \text{sonst} \end{cases}$$

$$T_{euler}(d) = \begin{cases} T_{fak}(d) + T_{euler}(d-1) + 3, & \text{wenn } d \geq 1 \\ 0, & \text{sonst} \end{cases}$$

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:

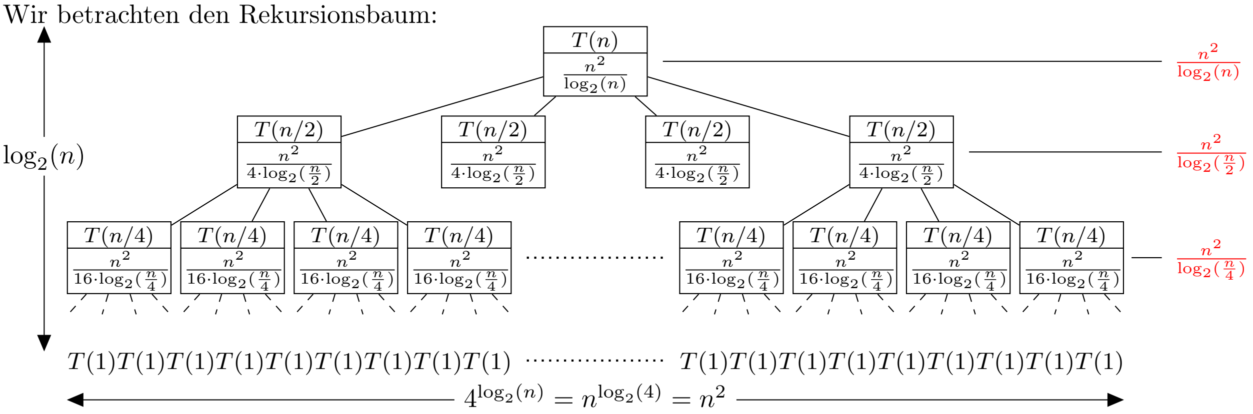
i) $\forall n \geq 1.\ \log(n) < H_n$

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

Lösung:

Hinweis:
Das Mastertheorem ist nicht anwendbar, da $\frac{n^2}{\log_2(n)} \notin \mathcal{O}(n^{2-\epsilon})$, $\frac{n^2}{\log_2(n)} \notin \Theta(n^2)$ *und* $\frac{n^2}{\log_2(n)} \notin \Omega(n^{2+\epsilon})$.

Wir betrachten den Rekursionsbaum:



Aus dem Rekursionsbaum leiten wir folgende Abschätzung ab:

$$\begin{aligned}
T(n) &= \left(\sum_{i=0}^{\log_2(n)-1} \frac{n^2}{\log_2(\frac{n}{2^i})}\right) + n^2 \\
&= n^2 \cdot \left(\sum_{i=0}^{\log_2(n)-1} \frac{1}{\log_2(n)-\log_2(2^i)}\right) + n^2 \\
&= n^2 \cdot \left(\sum_{i=0}^{\log_2(n)-1} \frac{1}{\log_2(n)-i}\right) + n^2 \\
&= n^2 \cdot \left(\sum_{i=1}^{\log_2(n)} \frac{1}{i}\right) + n^2 && \text{(Summe gedreht)} \\
&\simeq n^2 \cdot \log(\log_2(n)) + n^2 && \text{(mit wachsendem } n\text{)}
\end{aligned}$$

Daher stellen wir folgende Vermutung auf:

$$T(n) \in \mathcal{O}(n^2 \cdot \log(\log_2(n)))$$

Behauptung: $T(n) \in \mathcal{O}(n^2 \cdot \log(\log_2(n)))$

Beweis:
Wir müssen zeigen, dass:

$$\exists c_1 > 0.\ \exists n_0 \geq 0.\ \forall n \geq n_0.\ T(n) \leq c_1 \cdot n^2 \cdot \log(\log_2(n)))$$

Wähle $n_0 = 4$ und $c_1 = 4$.

Induktionsanfang: $T(4) = 40 \leq 4 \cdot 4^2 \cdot \log(\log_2(4)) \approx 4 \cdot 11,09$
Induktionsvoraussetzung: $\forall n_0 \leq m < n.\ T(m) \leq c_1 \cdot m^2 \cdot \log(\log_2(m))$
Induktionsschluss:

$$\begin{aligned}
T(n) &= 4 \cdot T(\tfrac{n}{2}) + \frac{n^2}{\log_2(n)} \\
&\leq 4 \cdot c_1 \cdot \frac{n^2}{4} \cdot \log(\log_2(\tfrac{n}{2})) + \frac{n^2}{\log_2(n)} && \Big|\ \text{Induktionsvoraussetzung} \\
&= c_1 \cdot n^2 \cdot \log(\log_2(n) - \log_2(2)) + \frac{n^2}{\log_2(n)} \\
&= c_1 \cdot n^2 \cdot \log(\log_2(n) - 1) + \frac{n^2}{\log_2(n)} \\
&< c_1 \cdot n^2 \cdot \left(\log(\log_2(n)) - \frac{1}{\log_2(n)}\right) + \frac{n^2}{\log_2(n)} && \Big|\ \text{(i)} \Rightarrow \log(k-1) < H_{k-1} - H_k + \log(k) = -\tfrac{1}{k} + \log(k) \\
&\leq c_1 \cdot n^2 \cdot \log(\log_2(n)) - \frac{c_1 \cdot n^2}{\log_2(n)} + \frac{n^2}{\log_2(n)} \\
&= c_1 \cdot n^2 \cdot \log(\log_2(n)) + \underbrace{(1-c_1) \cdot (n^2/\log_2(n))}_{<0 \textit{ für } c_1=4 \text{ und } n \geq n_0=2} \\
&\leq c_1 \cdot n^2 \cdot \log(\log_2(n))
\end{aligned}$$

$\square$

Aufgabe 4 (Substitutionsmethode): (8 Punkte)

Geben Sie zur Rekursionsgleichung

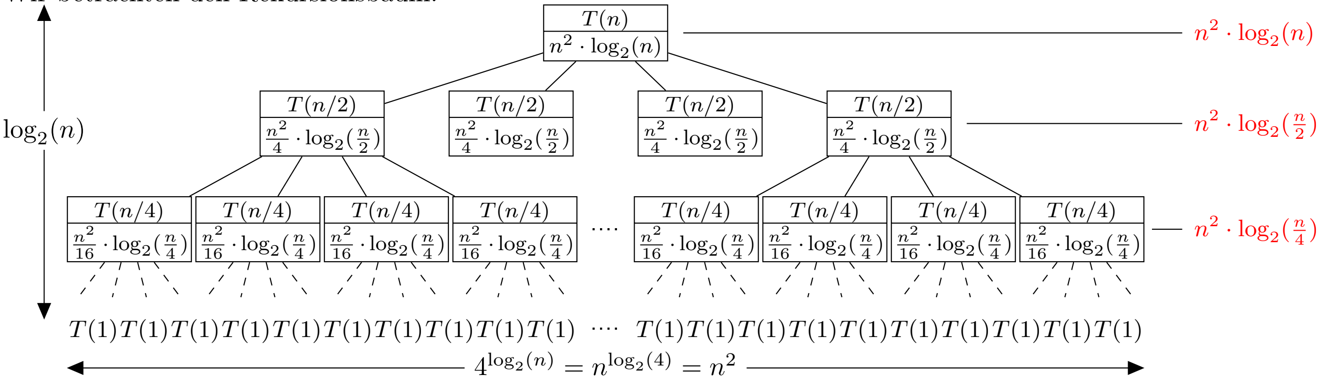
$$T(n) = \begin{cases} 4 \cdot T(\frac{n}{2}) + n^2 \cdot \log_2(n), & \text{falls } n > 1 \\ 1, & \text{sonst} \end{cases}$$

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

Lösung:
Hinweis:

Das Mastertheorem ist nicht anwendbar, da $n^2 \cdot \log_2(n) \notin \mathcal{O}(n^{2-\epsilon})$, $n^2 \cdot \log_2(n) \notin \Theta(n^2)$ *und* $n^2 \cdot \log_2(n) \notin \Omega(n^{2+\epsilon})$.

Wir betrachten den Rekursionsbaum:



Aus dem Rekursionsbaum leiten wir folgende Abschätzung ab:

$$\begin{aligned}
T(n) &= \left(\sum_{i=0}^{\log_2(n)-1} n^2 \cdot \log_2\left(\tfrac{n}{2^i}\right)\right) + n^2 \\
&= n^2 \cdot \left(\sum_{i=0}^{\log_2(n)-1} \log_2(n) - \log_2(2^i)\right) + n^2 \\
&= n^2 \cdot \left(\sum_{i=0}^{\log_2(n)-1} \log_2(n) - i\right) + n^2 \\
&= n^2 \cdot \left(\sum_{i=0}^{\log_2(n)-1} \log_2(n) - \sum_{i=0}^{\log_2(n)-1} i\right) + n^2 \\
&= n^2 \cdot \left((\log_2(n))^2 - \tfrac{\log_2(n)\cdot(\log_2(n)-1)}{2}\right) + n^2 \\
&= n^2 \cdot \tfrac{(\log_2(n))^2}{2} + n^2 \cdot \tfrac{\log_2(n)}{2} + n^2
\end{aligned}$$

Daher stellen wir folgende Vermutung auf:

$$T(n) \in \Theta(n^2 \cdot (\log_2(n))^2)$$

Wir zeigen zunächst:

Behauptung: $T(n) \in \mathcal{O}(n^2 \cdot (\log_2(n))^2)$

Beweis:
Wir müssen zeigen, dass:

$$\exists c_1 > 0.\ \exists n_0 \geq 0.\ \forall n \geq n_0.\ T(n) \leq c_1 \cdot n^2 \cdot (\log_2(n))^2$$

Wähle $n_0 = 2$ und $c_1 = 2$.

<u>Induktionsanfang</u>: $T(2) = 8 \leq 2 \cdot 2^2 \cdot (\log_2(2))^2$
<u>Induktionsvoraussetzung</u>: $\forall n_0 \leq m < n.\ T(m) \leq c_1 \cdot m^2 \cdot (\log_2(m))^2$
<u>Induktionsschluss</u>:

$$\begin{aligned}
T(n) &= 4 \cdot T\left(\tfrac{n}{2}\right) + n^2 \cdot \log_2(n) \\
&\leq 4 \cdot c_1 \cdot \tfrac{n^2}{4}\left(\log_2\left(\tfrac{n}{2}\right)\right)^2 + n^2 \cdot \log_2(n) && \text{(Induktionsvoraussetzung)} \\
&= c_1 \cdot n^2 \cdot (\log_2(n) - \log_2(2))^2 + n^2 \cdot \log_2(n) \\
&= c_1 \cdot n^2 \cdot ((\log_2(n))^2 - 2\log_2(n) + 1) + n^2 \cdot \log_2(n) \\
&= c_1 \cdot n^2 \cdot (\log_2(n))^2 - 2 \cdot c_1 \cdot n^2 \cdot \log_2(n) + c_1 \cdot n^2 + n^2 \cdot \log_2(n) \\
&= c_1 \cdot n^2 \cdot (\log_2(n))^2 \quad + (1 - 2 \cdot c_1) \cdot n^2 \cdot \log_2(n) + c_1 \cdot n^2 \\
&= c_1 \cdot n^2 \cdot (\log_2(n))^2 \quad + n^2 \cdot \underbrace{((1 - 2 \cdot c_1) \cdot \log_2(n) + c_1)}_{<0 \text{ für } c_1 = 2 \text{ und } n \geq n_0 = 2} \\
&\leq c_1 \cdot n^2 \cdot (\log_2(n))^2
\end{aligned}$$

$\square$

Analog zeigen wir:

Behauptung: $T(n) \in \Omega(n^2 \cdot (\log_2(n))^2)$

Beweis:
Wir müssen zeigen, dass:

$$\exists c_2 > 0.\ \exists n_0 \geq 0.\ \forall n \geq n_0.\ T(n) \geq c_1 \cdot n^2 \cdot (\log_2(n))^2$$

Wähle $n_0 = 2$ und $c_2 = \frac{1}{4}$.

Induktionsanfang: $T(2) = 8 \geq \frac{1}{4} \cdot 2^2 \cdot (\log_2(2))^2$
Induktionsvoraussetzung: $\forall n_0 \leq m < n.\ T(m) \geq c_2 \cdot m^2 \cdot (\log_2(m))^2$
Induktionsschluss:

$$\begin{aligned}
T(n) &= 4 \cdot T(\tfrac{n}{2}) + n^2 \cdot \log_2(n) \\
&\geq 4 \cdot c_2 \cdot \tfrac{n^2}{4}(\log_2(\tfrac{n}{2}))^2 + n^2 \cdot \log_2(n) && \text{(Induktionsvoraussetzung)} \\
&= \ldots \\
&= c_2 \cdot n^2 \cdot (\log_2(n))^2 \quad + (1 - 2 \cdot c_2) \cdot n^2 \cdot \log_2(n) + c_2 \cdot n^2 \\
&= c_2 \cdot n^2 \cdot (\log_2(n))^2 \quad + n^2 \cdot \underbrace{((1 - 2 \cdot c_2) \cdot \log_2(n) + c_2)}_{>0 \text{ für } c_2 = \frac{1}{4} \text{ und } n \geq n_0 = 2} \\
&\geq c_2 \cdot n^2 \cdot (\log_2(n))^2
\end{aligned}$$

$\square$

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 $n > 1$

Lösung:

a) Behauptung: $T(n) \in \Theta(n^3)$

Beweis:
Wir haben $a = 27$, $b = 3$ und $f(n) = n^2 \cdot \log(n)$. Damit ergibt sich $E = \log(27)/\log(3) = 3$. Wir zeigen $n^2 \cdot \log(n) \in \mathcal{O}(n^{3-\varepsilon})$ für ein $\varepsilon > 0$. Wir haben $\lim_{n\to\infty} \frac{n^2 \cdot \log(n)}{n^{3-\varepsilon}} = \lim_{n\to\infty} \frac{\log(n)}{n^{1-\varepsilon}} \overset{\text{L'Hôpital}}{=} \lim_{n\to\infty} \frac{1}{(1-\varepsilon)\cdot n^{1-\varepsilon}} = 0$ für $0 < \varepsilon < 1$. Damit gilt nach dem ersten Fall des Mastertheorems $T(n) \in \Theta(n^3)$.

$\square$

b) Behauptung: $T(n) \in \Theta(n^{\frac{5}{2}})$

Beweis:
Wir haben $a = 16$, $b = 4$ und $f(n) = \frac{n^3}{\sqrt{n}}$. Damit ergibt sich $E = \log(16)/\log(4) = 2$. Wir zeigen $\frac{n^3}{\sqrt{n}} \in \Omega(n^{2+\varepsilon})$ für ein $\varepsilon > 0$. Wir haben $\lim_{n\to\infty} \frac{\frac{n^3}{\sqrt{n}}}{n^{2+\varepsilon}} = \lim_{n\to\infty} n^{\frac{1}{2}-\varepsilon} = \infty$ für $0 < \varepsilon < \frac{1}{2}$. Außerdem haben wir $16 \cdot \frac{(\frac{n}{4})^3}{\sqrt{\frac{n}{4}}} = \frac{1}{2} \cdot \frac{n^3}{\sqrt{n}}$. Damit gilt nach dem dritten Fall des Mastertheorems $T(n) \in \Theta(n^{\frac{5}{2}})$. $\square$

c) Behauptung: Das Mastertheorem ist nicht anwendbar.

Beweis:
Wir haben $a = 2$, $b = 2$ und $f(n) = \frac{n}{\log(n)}$. Damit ergibt sich $E = \log(2)/\log(2) = 1$. Für den ersten Fall müssten wir $\frac{n}{\log(n)} \in \mathcal{O}(n^{1-\varepsilon})$ für ein $\varepsilon > 0$ haben. Es gilt aber $\lim_{n\to\infty} \frac{\frac{n}{\log(n)}}{n^{1-\varepsilon}} = \lim_{n\to\infty} \frac{n^\varepsilon}{\log(n)} \overset{\text{L'Hôpital}}{=} \lim_{n\to\infty} \varepsilon \cdot n^\varepsilon = \infty$ für alle $\varepsilon > 0$. Für den zweiten Fall müssten wir $\frac{n}{\log(n)} \in \Theta(n)$ haben, aber es gilt $\lim_{n\to\infty} \frac{\frac{n}{\log(n)}}{n} = \lim_{n\to\infty} \frac{1}{\log(n)} = 0$. Für den dritten Fall müssten wir $\frac{n}{\log(n)} \in \Omega(n^{1+\varepsilon})$ für ein $\varepsilon > 0$ haben. Es gilt aber $\lim_{n\to\infty} \frac{\frac{n}{\log(n)}}{n^{1+\varepsilon}} = \lim_{n\to\infty} \frac{1}{n^\varepsilon \cdot \log(n)} = 0$ für alle $\varepsilon > 0$. $\square$

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 $n > 1$

Lösung:

---

a) Behauptung: $T(n) \in \Theta(\sqrt{n} \cdot \log(n))$

Beweis:
Wir haben $a = 2$, $b = 4$ und $f(n) = \sqrt{n}$. Damit ergibt sich $E = \log(2)/\log(4) = \frac{1}{2}$. Wir zeigen $\sqrt{n} \in \Theta(n^{\frac{1}{2}})$. Wir haben $\lim_{n\to\infty} \frac{\sqrt{n}}{n^{\frac{1}{2}}} = 1$. Damit gilt nach dem zweiten Fall des Mastertheorems $T(n) \in \Theta(\sqrt{n} \cdot \log(n))$. $\square$

b) Behauptung: Das Mastertheorem ist nicht anwendbar.

Beweis:
Wir haben $a = 25$, $b = 5$ und $f(n) = (n \cdot \log(n))^2$. Damit ergibt sich $E = \log(25)/\log(5) = 2$. Für den ersten Fall müssten wir $(n \cdot \log(n))^2 \in \mathcal{O}(n^{2-\varepsilon})$ für ein $\varepsilon > 0$ haben. Es gilt aber $\lim_{n\to\infty} \frac{(n \cdot \log(n))^2}{n^{2-\varepsilon}} = \lim_{n\to\infty} (\log(n))^2 \cdot n^\varepsilon = \infty$ für alle $\varepsilon > 0$. Für den zweiten Fall müssten wir $(n \cdot \log(n))^2 \in \Theta(n^2)$ haben, aber es gilt $\lim_{n\to\infty} \frac{(n \cdot \log(n))^2}{n^2} = \lim_{n\to\infty} (\log(n))^2 = \infty$. Für den dritten Fall müssten wir $(n \cdot \log(n))^2 \in \Omega(n^{2+\varepsilon})$ für ein $\varepsilon > 0$ haben. Es gilt aber $\lim_{n\to\infty} \frac{(n \cdot \log(n))^2}{n^{2+\varepsilon}} = \lim_{n\to\infty} \frac{(\log(n))^2}{n^\varepsilon} \overset{\text{L'Hôpital}}{=} \lim_{n\to\infty} \frac{2 \cdot \log(n)}{\varepsilon \cdot n^\varepsilon} \overset{\text{L'Hôpital}}{=} \lim_{n\to\infty} \frac{2}{\varepsilon^2 \cdot n^\varepsilon} = 0$ für alle $\varepsilon > 0$. $\square$

c) Behauptung: $T(n) \in \Theta(\frac{n}{\log(n)})$

Beweis:
Wir haben $a = 3$, $b = 9$ und $f(n) = \frac{n}{\log(n)}$. Damit ergibt sich $E = \log(3)/\log(9) = \frac{1}{2}$. Wir zeigen $\frac{n}{\log(n)} \in \Omega(n^{\frac{1}{2}+\varepsilon})$ für ein $\varepsilon > 0$. Wir haben $\lim_{n\to\infty} \frac{\frac{n}{\log(n)}}{n^{\frac{1}{2}+\varepsilon}} = \lim_{n\to\infty} \frac{n^{\frac{1}{2}-\varepsilon}}{\log(n)} \overset{\text{L'Hôpital}}{=} \lim_{n\to\infty} (\frac{1}{2}-\varepsilon) \cdot n^{\frac{1}{2}-\varepsilon} = \infty$ für $0 < \varepsilon < \frac{1}{2}$. Außerdem haben wir $3 \cdot \frac{\frac{n}{9}}{\log(\frac{n}{9})} = \frac{1}{3} \cdot \frac{n}{\log(n)-\log(9)} < \frac{1}{2} \cdot \frac{n}{\log(n)}$ für $n > 729$. Denn: $\frac{1}{3} \cdot \frac{n}{\log(n)-\log(9)} < \frac{1}{2} \cdot \frac{n}{\log(n)} \overset{n>729}{\iff} 3 \cdot \log(9) < \log(n) \iff n > e^{3\cdot\log(9)} = 729$. Damit gilt nach dem dritten Fall des Mastertheorems $T(n) \in \Theta(\frac{n}{\log(n)})$. $\square$