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:

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

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.

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 <u>Substitutionsmethode</u>, 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:

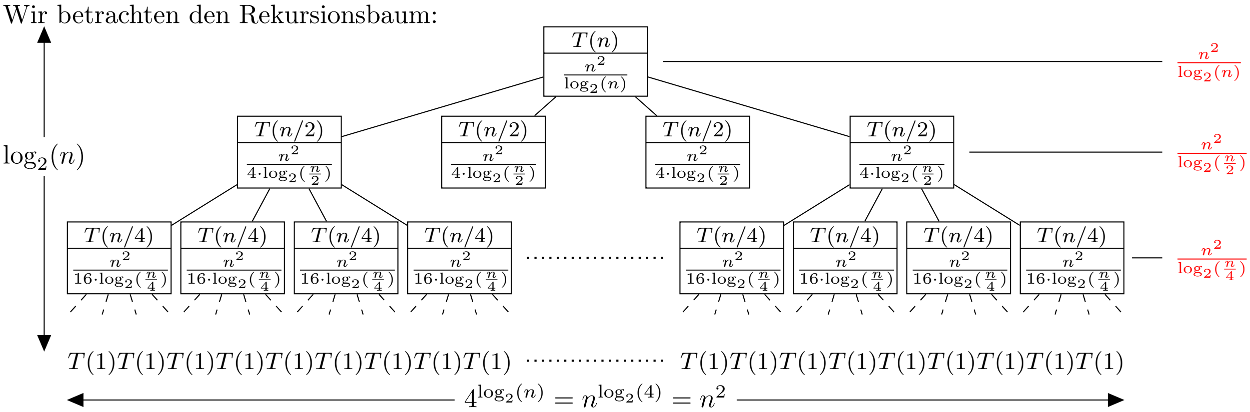
i) $\forall n \geq 1.\ \log(n) < H_n$

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:

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

$\square$

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 $\varepsilon > 0$. $\square$