Lösungsvorschläge zur 1. Übung

Aufgabe 1

Bei der Implementation des *MergeSort*–Algorithmus gab es hauptsächlich Probleme durch die Prozedur *Merge*. In ihr sollten zwei sortierte Bereiche eines Feldes zu einem sortierten Bereich zusammengefaßt werden. Viele haben versucht, der Reihe nach die Elemente aus dem ersten Bereich in den zweiten Bereich einzusortieren. Dies ist jedoch nicht erlaubt, da sonst die Prozedur *Merge* die Laufzeit $O(N^2)$ hätte. Ein Listing als Lösungsvorschlag kommt nächste Woche.

Aufgabe 3

Bei der Aufgabe 3 sind die Probleme hauptsächlich durch fehlende Definitionen entstanden. Man könnte folgende Defintionen einführen:

$$\begin{aligned}
cO(f(n)) &:= \{cg(n) : g(n) \in O(f(n))\} \\
f(n)O(g(n)) &:= \{f(n)h(n) : h(n) \in O(g(n))\} \\
O(f(n))O(g(n)) &:= \{p(n)q(n) : p(n) \in O(f(n)) \wedge q(n) \in O(g(n))\} \\
O(O(f(n))) &:= \bigcup_{g(n)\in O(f(n))} O(g(n))
\end{aligned}$$

Damit ergeben sich folgende Lösungen:

zu a)

Laut Analysis gilt:

$$\bigwedge_{n,q,p\geq 1 \wedge p\leq q} n^p \leq n^q$$

also auch für ein beliebiges $c \geq 1$:

$$\bigwedge_{n,q,p\leq 1 \wedge p\leq q} n^p \leq cn^q$$

Demnach in ist $n^p \in O(n^q)$ für $p \leq q$ mit $n_0 := 1$ und beliebigem $c \geq 1$.

zu b)

Es gilt für alle n und alle $c \geq 1$:

$$f(n) \leq cf(n)$$

Also ist $f(n) \in O(f(n))$ mit $n_0 := 1$ und beliebigem $c > 1$.

zu d)

$$cO(f(n)) = \left\{ \underbrace{cg(n)}_{=:h(n)} : \bigvee_{d>0, n_0\in\mathbf{N}} \bigwedge_{n\geq n_0} g(n) \leq df(n) \right\}$$

$$
\begin{aligned}
&= \left\{h(n): \bigvee_{d>0, n_0 \in \mathbf{N}} \bigwedge_{n \geq n_0} h(n) \leq \underbrace{cd}_{=:\tilde{d}} f(n)\right\} \\
&= \left\{h(n): \bigvee_{\tilde{d}>0, n_0 \in \mathbf{N}} \bigwedge_{n \geq n_0} h(n) \leq \tilde{d} f(n)\right\} \\
&= O(f(n))
\end{aligned}
$$

zu e)

Sei $p(n) \in O(O(f(n)))$ beliebig aber fest gegeben. Es ist jetzt zu zeigen, daß $p(n) \in O(f(n))$ ist. Laut Definition existiert ein $g(n) \in O(f(n))$, so daß $p(n) \in O(g(n))$. Das ist aber gleichbedeutend damit, daß $n_0, m_0 \in \mathbf{N}$ und $c, d > 0$ existieren, so daß gilt:

$$
\bigwedge_{n \geq n_0} g(n) \leq cf(n) \qquad \text{sowie} \qquad \bigwedge_{n \geq m_0} p(n) \leq dg(n)
$$

Dann gilt aber auch (durch Einsetzen der ersten Ungleichung in die zweite):

$$
\bigwedge_{n \geq \underbrace{\max\{n_0, m_0\}}_{=:\tilde{m_0}}} p(n) \leq \underbrace{dc}_{=:\tilde{d}} f(n)
$$

Also ist $p(n) \in O(f(n))$.

zu f)

Sei $r(n) \in O(f(n))O(p(n))$ beliebig aber fest gegeben. Zu zeigen ist, daß $r(n) \in O(f(n)g(n))$ ist. Laut Definition existieren $p(n) \in O(f(n))$ und $q(n) \in O(g(n))$ sowie $n_0, m_0 \in \mathbf{N}$ und $c, d > 0$ mit

$$
r(n) = p(n)q(n) \qquad \text{und} \qquad \bigwedge_{n \geq n_0} p(n) \leq cf(n) \qquad \text{und} \qquad \bigwedge_{n \geq m_0} q(n) \leq dg(n)
$$

Dann gilt aber auch (durch Multiplikation der beiden Ungleichungen):

$$
\bigwedge_{n \geq \underbrace{\max\{n_0, m_0\}}_{=:\tilde{m_0}}} r(n) = p(n)q(n) \leq \underbrace{cd}_{=:\tilde{d}} f(n)g(n)
$$

Also ist $r(n) \in O(f(n)g(n))$.

zu g)

Sei $p(n) \in O(f(n)g(n))$ beliebig aber fest gegeben. Zu zeigen ist, daß $p(n) \in f(n)O(g(n))$. Nach Definition existieren $n_0 \in \mathbf{N}$ sowie $c > 0$ mit

$$
\bigwedge_{n \geq n_0} p(n) \leq \underbrace{cf(n)g(n)}_{=f(n)cg(n)}
$$

Also $p(n) \in f(n)O(g(n))$ mit gleichem n_0 und gleichem c.

Aufgabe 4

Für die Lösung der Aufgabe 4 braucht man folgende Überlegungen:

- Da die Aussagen über obere und untere Schranken erst ab großen n sinnvoll sind, kann das Abrunden $\lfloor \quad \rfloor$ vernachlässigt werden, denn der dadurch entstehende *relative Fehler* wird immer kleiner.
- Aus gleichem Grund braucht man bei einer Funktion $f(n)$, die aus einer Summe besteht, nur den denjenigen Summanden zu betrachten, der am schnellsten wächst.

Seien nun zwei Funktionen $f(n)$ und $g(n)$ gegeben. Für die Ermittlung der oberen und unteren Schranke betrachtet man folgenden Grenzübergang:

$$l := \lim_{n\to\infty} \frac{f(n)}{g(n)}$$

Es gibt nun drei Fälle:

1. Wenn $l = \infty$, so stellt $f(n)$ eine obere Schranke für $g(n)$ dar. Es gilt:
$$g(n) = O(f(n)) \qquad \text{und} \qquad f(n) = \Omega(g(n))$$
2. Wenn $l = 0$, so stellt $f(n)$ eine untere Schranke für $g(n)$ dar. Es gilt:
$$f(n) = O(g(n)) \qquad \text{und} \qquad g(n) = \Omega(f(n))$$
3. Wenn $0 < l < \infty$, so stellt $f(n)$ sowohl eine untere Schranke als auch eine obere Schranke für $g(n)$ dar. Es gilt:
$$f(n) = \Theta(g(n)) \qquad \text{und} \qquad g(n) = \Theta(f(n))$$
Das so gefundene l dient dann direkt zur Abschätzung. Es existieren n_0 und m_0 mit:
$$\bigwedge_{n\geq n_0} f(n) \leq 2lg(n) \qquad \text{und} \qquad \bigwedge_{n\geq m_0} f(n) \geq \frac{1}{2}lg(n)$$

zu a)

$$\frac{f(n)}{g(n)} = \frac{\sqrt{n}}{1000n} = \frac{1}{1000\sqrt{n}} \stackrel{n\to\infty}{\longrightarrow} 0$$

Also Fall 2.

zu b)

$$\frac{f(n)}{g(n)} = \frac{\log_{10} n}{\log_2 n} = \frac{\frac{\log_2 n}{\log_2 10}}{\log_2 n} = \frac{1}{\log_2 10} =: l$$

Also Fall 3.

zu d)

$$\frac{f(n)}{g(n)} = \frac{n^2}{n\log_2 n} = \frac{n}{\log_2 n} \stackrel{n\to\infty}{\longrightarrow} \infty$$

Also Fall 1.