Institut für Informatik
der Universität München
Prof. Dr. M. Hofmann
Dr. J. Johannsen, Dr. M. Lange

WS 03/04

Übungen zur Vorlesung
Informatik I

Musterlösungen zu Blatt 13

Lösung zu Aufgabe S-47:
`rev_naive`: Laufzeit $O(n^2)$. Wichtig ist zu erkennen, dass die Operation `@` n Schritte braucht, wenn n die Anzahl der Elemente des linken Arguments von `@` ist. Somit gilt für die Laufzeit $T(n)$ von `rev_naive`:

$$T(0) = 1$$
$$T(n+1) = T(n) + n + 1$$

Durch Einsetzen z.B. kann man zeigen, dass $T(n) = O(n^2)$ eine Lösung für dieses Gleichungssystem ist.

`rev_clever`: Im Unterschied zu `rev_naive` gilt hier $T(n+1) = T(n)+1$, da bei jedem rekursiven Aufruf lediglich ein `::` ausgeführt werden muss. Einsetzen zeigt, dass $T(n) = O(n)$ eine Lösung ist.

Lösung zu Aufgabe S-48:
a) Sei n die Anzahl der Knoten in dem Baum, der an die Funktion übergeben wird.

b) Die Laufzeit hängt noch von der Frage ab, ob der Pfad in dem Baum vorhanden ist oder nicht, sowie – falls er es ist – von der genauen Position im Baum. Falls er nicht vorhanden ist, dann hängt die Laufzeit noch von der Position des ersten Elementes in der Liste ab, welches kein Knoten im Baum ist.

Der best case liegt vor, wenn der Pfad nicht im Baum vorhanden ist und das erste Element der Liste ungleich der Wurzel des Baums ist. Der worst case liegt vor, wenn entweder der Pfad nicht vorhanden ist, aber nur das letzte Element der Liste nicht mehr in den Baum passt, oder wenn der Pfad ganz rechts im Baum vorhanden ist.

Unterschiedliche best und worst case Szenarien ergeben sich dadurch, dass das `||` nicht-strikt ausgewertet wird. Somit muss evtl. nicht der gesamte Baum nach dem Pfad durchsucht werden.

c) Best case wie oben beschrieben in $O(1)$, denn nur die Wurzel und das erste Element der Liste müssen verglichen werden. Worst case wie oben beschrieben $O(n)$, denn u.U. muss der gesamte Baum durchsucht werden.

Lösung zu Aufgabe S-49:
a) Für alle $n \geq 1$ gilt $n! \leq n^n$, daraus erhält man durch logarithmieren

$$\log n! \leq \log n^n = n \log n \ ,$$

also gilt die Aussage $\log n! = O(n \log n)$ mit $N = 1$ und $c = 1$.

b) Es ist

$$\begin{aligned}\log n! &= \log n + \log(n-1) + \ldots + \log 2 \\ &\geq \log n + \log(n-1) + \ldots + \log n/2 \\ &\geq \log n/2 + \log n/2 + \ldots + \log n/2 \\ &= n/2 \log n/2 = \frac{1}{2} n(\log n - \log 2) \\ &\geq \frac{1}{4} n \log n\end{aligned}$$

wobei die letzte Ungleichung für alle n gilt mit $\frac{1}{2}\log 2 \leq \frac{1}{4}\log n$, also für $n \geq 4$.

Damit haben wir $n \log n \leq 4 \log n!$ für alle $n \geq 4$.

c) Zunächst ist $2^x > x$ für alle $0 \leq x \in \mathbb{R}$. Wir setzen δn für x ein, wobei δ eine noch zu bestimmende Konstante ist, also

$$2^{\delta n} > \delta n\ .$$

Wählen wir nun $\delta = \epsilon/2$, so ist

$$2^{\epsilon n} > 2^{\delta n + \log(1/\delta)} = \frac{1}{\delta} 2^{\delta n} > n$$

für alle n mit $\delta n + \log(1/\delta) < \epsilon n$. Dies ist der Fall, wenn $\epsilon n/2 > \log(2/\epsilon)$, also für $n > 2/\epsilon \log 2/\epsilon$. Substitution von $\log n$ für n liefert

$$\log n < 2^{\epsilon \log n} = n^{\epsilon}$$

und Multiplikation mit n ergibt die Behauptung $n \log n = O(n^{\epsilon})$ mit $c = 1$ und $N = \lceil 2/\epsilon \log 2/\epsilon \rceil$.

Lösung zu Aufgabe S-50:

a) Für alle $n \geq 0$ ist $f(n) \leq f(n)$, also gilt die Behauptung mit $N = 0$ und $c = 1$.

b) Nach Vorraussetzung ist $f(n) \leq c_1 g(n)$ für alle $n \geq N_1$ und $g(n) \leq c_2 h(n)$ für $n \geq N_2$. Sei $N := \max(N_1, N_2)$, dann gilt für $n \geq N$

$$f(n) \leq c_1 g(n) \leq c_1 c_2 h(n)$$

also gilt die Aussage mit N wie gewählt und $c = c_1 c_2$.

c) Sei $n \geq 0$ gegeben. Falls $f(n) \leq g(n)$ ist, so ist $\max(f(n), g(n)) = g(n)$, und daher

$$f(n) + g(n) \leq 2g(n) \leq 2\max(f(n), g(n))\ ,$$

sonst ist $f(n) > g(n)$, also $\max(f(n), g(n)) = f(n)$, und es gilt

$$f(n) + g(n) \leq 2f(n) \leq 2\max(f(n), g(n))\ .$$

In jedem Fall gilt die Behauptung mit $N = 0$ und $c = 2$.