10. Übungsblatt zu Programmierung 1, WS 2010/11

Prof. Dr. Gert Smolka, Christian Doczkal, M.Sc.
www.ps.uni-sb.de/courses/prog-ws10/

Lesen Sie im Buch: Kapitel 11

Aufgabe 11.2 Geben Sie für die folgende Prozedur eine Größenfunktion und die entsprechende Laufzeitfunktion an:

$$\begin{aligned}&\mathit{revi} : \mathcal{L}(X) \times \mathcal{L}(X) \to \mathcal{L}(X)\\ &\mathit{revi}(xs,\ \mathit{nil}) = xs\\ &\mathit{revi}(xs,\ y :: yr) = \mathit{revi}(y :: xs,\ yr)\end{aligned}$$

Aufgabe 11.6 Betrachten Sie die Prozedur *insert*, die ein neues Element in eine Liste einfügt (Sortieren durch Einfügen, § 5.1):

$$\begin{aligned}&\mathit{insert} : \mathbb{Z} \times \mathcal{L}(\mathbb{Z}) \to \mathcal{L}(\mathbb{Z})\\ &\mathit{insert}(x, \mathit{nil}) = [x]\\ &\mathit{insert}(x, y :: yr) = \text{if } x \le y \text{ then } x :: y :: yr \text{ else } y :: \mathit{insert}(x, yr)\end{aligned}$$

a) Wählen Sie eine Größenfunktion für *insert*.
b) Geben Sie für die Größe 4 ein Argument mit minimaler Laufzeit und ein Argument mit maximaler Laufzeit an (in Bezug auf Argumente der Größe 4).
c) Welche minimale und maximale Laufzeit hat *insert* für Argumente der Größe n?
d) Geben Sie die Laufzeitfunktion von *insert* an.

Aufgabe 11.8 Betrachten Sie die folgende Prozedur:

$$\begin{aligned}&p : \mathbb{N} \times \mathbb{N} \to \mathbb{N}\\ &p(0, k) = 0\\ &p(n, k) = p(n-1, 0) + \cdots + p(n-1, k) \quad \text{für } n > 0\end{aligned}$$

a) Geben Sie die Laufzeit von p für das Argument $(1, k)$ an.
b) Offensichtlich ist $\lambda\,(n,k).\ n$ eine natürliche Terminierungsfunktion für p. Machen Sie sich klar, dass $\lambda\,(n,k).\ n$ aber keine Größenfunktion für p ist.
c) Geben Sie eine Größenfunktion für p an.

Aufgabe 11.9 Geben Sie eine rekursive Beschreibung der Laufzeitfunktion der Prozedur @ gemäß der in § 11.2.1 angegebenen Größenfunktion an.

Aufgabe 11.16 Geben Sie eine baumrekursive Prozedur $\mathbb{N} \to \mathbb{N}$ an, deren Komplexität konstant ist. Hinweis: Formulieren Sie die Prozedur so, dass sie ab einer bestimmten Argumentgröße nicht mehr rekurriert. Das Beispiel zeigt, dass man pathologische Prozeduren konstanter Komplexität konstruieren kann, die für praktisch relevante Argumente sehr hohe Laufzeiten haben.

Aufgabe 11.18 Geben Sie Komplexitäten der durch die folgenden Gleichungen rekursiv definierten Funktionen $f \in \mathbb{N} \to \mathbb{N}$ an:

a) $f\,n = \text{if } n = 0 \text{ then } 0 \text{ else } f(n-1)$
b) $f\,n = \text{if } n = 0 \text{ then } 1 \text{ else } f(n-1)$
c) $f\,n = \text{if } n = 0 \text{ then } 0 \text{ else } f(n-1) + 1$

Aufgabe 11.19 Geben Sie Komplexitäten der durch die folgenden Gleichungen definierten Prozeduren $f : \mathbb{N} \to \mathbb{N}$ an:

a) $f\,n = \text{if } n = 0 \text{ then } 0 \text{ else } f(n-1)$
b) $f\,n = \text{if } n = 0 \text{ then } 1 \text{ else } f(n-1)$
c) $f\,n = \text{if } n = 0 \text{ then } 0 \text{ else } f(n-1) + 1$

Vergleichen Sie Ihre Ergebnisse mit denen aus Aufgabe 11.18.

Aufgabe 11.24 Geben Sie für eine beliebige Größe $n \in \mathbb{N}$ Argumente an, für die *isort* minimale beziehungsweise maximale Laufzeit hat.

Aufgabe 11.25 Sie sollen die Best- und die Worst-Case-Laufzeit der folgenden Prozedur bestimmen:

$gcd : \mathbb{N}_+ \times \mathbb{N}_+ \to \mathbb{N}_+$
$gcd(x,x) = x$
$gcd(x,y) = gcd(x-y,\, y) \quad \text{für } x > y$
$gcd(x,y) = gcd(x,\, y-x) \quad \text{für } x < y$

Dabei soll die Größenfunktion $\lambda\,(x,y).\ \max\{x,y\}$ zugrunde gelegt werden.

a) Geben Sie Argumente für die Größe $n = 4$ an, für die die Laufzeit minimal beziehungsweise maximal ist. Geben Sie die jeweiligen Rekursionsfolgen an.
b) Geben Sie für eine beliebige Größe $n \geq 1$ Argumente an, für die die Laufzeit minimal beziehungsweise maximal ist.
c) Geben Sie die Best-Case-Laufzeitfunktion und deren Komplexität an.
d) Geben Sie die Worst-Case-Laufzeitfunktion und deren Komplexität an.

Aufgabe 11.29 Geben Sie die Komplexitäten der im Folgenden rekursiv definierten Funktionen $f \in \mathbb{N} \to \mathbb{R}$ möglichst einfach an.

a) $f\,n = \text{if } n < 3 \text{ then } 1 \text{ else } f(n-2) + 5$
b) $f\,n = \text{if } n < 30 \text{ then } 1 \text{ else } f(n-6) + 3n + 7n^2$
c) $f\,n = \text{if } n < 27 \text{ then } 1 \text{ else } f(n-7) + 3n^7 + 7n^3$

Aufgabe 11.30 Geben Sie möglichst einfache Prozeduren $\mathbb{N} \to \{0\}$ an, die für die Größenfunktion $\lambda\,n.\ n$ die Komplexitäten $O(1)$, $O(n)$, $O(n^2)$ und $O(n^3)$ haben. Für $O(1)$ und $O(n)$ sollen keine Nebenkosten anfallen. Für $O(n^2)$ und $O(n^3)$ sollen lineare beziehungsweise quadratische Nebenkosten anfallen.

Aufgabe 11.31 Geben Sie möglichst einfache Prozeduren $\mathbb{N} \to \{0\}$ mit den Komplexitäten $O(2^n)$ und $O(3^n)$ an.

Aufgabe 11.36 Geben Sie eine möglichst einfache Prozedur $\mathbb{N} \to \{0\}$ an, die für die Größenfunktion $\lambda\,n.\ n$ die Komplexität $O(\log n)$ hat.

Aufgabe 11.39 Geben Sie eine möglichst einfache Prozedur $\mathbb{N} \to \{0\}$ an, die für die Größenfunktion $\lambda\,n.\ n$ die Komplexität $O(n \cdot \log n)$ hat. Benutzen Sie eine Funktion, für die lineare Nebenkosten anfallen.