Petra Mutzel
Nicola Beume, Christian Bockermann, Christian Horoba,
Ingo Schulz, Dirk Sudholt, Christine Zarges

Sommersemester 2009

DAP2 Übung – Blatt 2

Ausgabe: 23. April, **Abgabe:** 30. April, 14:00 Uhr, **Block:** A

Aufgabe 2.1 (4 Punkte) Entfernen von Duplikaten aus einfach verketteten Listen und Feldern

Der ADT *Sequence* kann z. B. mithilfe einer einfach verketteten Liste oder eines Feldes realisiert werden.

Betrachte zunächst eine einfach verkettete Liste (nicht ringförmig). Nimm an, die Elemente der Liste wären aufsteigend sortiert. Gib den Pseudocode für die Funktion

$$\textsc{RemoveDuplicates}(\text{ListElement } \textit{head})$$

an. Diese erwartet einen Zeiger auf das erste Listenelement als Eingabe. Nach Ausführung der Funktion soll *head* immer noch auf den Listenanfang zeigen, doch die Liste soll keine Elemente mit gleichem Wert mehr enthalten. Kommentiere dein Vorgehen.

Beschreibe danach kurz (ohne Pseudocode), wo die wesentlichen Unterschiede liegen, wenn man der Funktion RemoveDuplicates ein Feld als Datenstruktur zugrunde legt.

Aufgabe 2.2 (4 Punkte) Analyse eines Sortierverfahrens

Betrachte das folgende Sortierverfahren namens Bubble-Sort:

function $\textsc{BubbleSort}(A)$
  for $i := n-1, \ldots, 1$ **do**
    for $j := 1 \ldots i$ **do**
      if $A[j] > A[j+1]$ **then**
        vertausche $A[j]$ und $A[j+1]$
      end if
    end for
  end for
end function

Wende den Algorithmus auf die folgende Eingabeinstanz an:

S, O, R, T, I, E, R, E, N.

Gib dabei für jeden Durchlauf der äußeren Schleife den Feldinhalt vor dem Durchlauf der inneren Schleife an. Stelle die Datenbewegungen in der inneren Schleife durch Pfeile vom Ursprungsort zum Ziel dar.

Wie viele Schlüsselvergleiche $C_{\text{worst}}(n)$ und $C_{\text{best}}(n)$ benötigt Bubble-Sort im Worst- bzw. Best-Case? Wie viele Datenbewegungen $M_{\text{worst}}(n)$ und $M_{\text{best}}(n)$ benötigt Bubble-Sort im Worst- bzw. Best-Case? Gib jeweils die genaue Anzahl und die asymptotische Wachstumsordnung $\Theta(\cdot)$ an.

Aufgabe 2.3 (4 Punkte) Beziehungen zwischen verschiedenen O-Notationen

Betrachte Funktionen $f, g, h \colon \mathbb{N} \to \mathbb{R}^+$. Beweise die folgenden Aussagen:

1. Wenn $f(n) = O(g(n))$ und $g(n) = O(h(n))$, dann $f(n) = O(h(n))$.
2. $o(g(n)) \cap \omega(g(n)) = \emptyset$.
3. $\max\{f(n), g(n)\} = O(f(n) + g(n))$.

Hinweis: Verwende die Definitionen aus der Vorlesung und bedenke, dass die Ausdrücke $O(g(n))$, $O(h(n))$, $o(g(n))$, $\omega(g(n))$ und $O(f(n) + g(n))$ *Mengen* von Funktionen bezeichnen.

Präsenzaufgabe 2.4 O-Notation für zusammengesetzte Funktionen

Betrachte die Funktion $f \colon \mathbb{N} \to \mathbb{R}^+$ mit

$$f(n) = \begin{cases} 2n^3 & \text{falls } n \text{ gerade,} \\ 5n^2 & \text{falls } n \text{ ungerade.} \end{cases}$$

Bestimme asymptotische Schranken $\Omega(\cdot)$ und $O(\cdot)$ für f und beweise sie formal.