Karlsruher Institut für Technologie

Algorithmen 1 – Sommersemester 2018
Zuverlässige Softwaresysteme im Kontext der Automobilindustrie
`http://verialg.iti.kit.edu`

Prof. Dr. Carsten Sinz – Markus Iser

Übungsblatt 2

Ausgabe: 02.05.2018 – 15:30
Abgabe: 09.05.2018 – 13:00

A Rekurrenzen

A.1 Obere Schranke beweisen (2 Punkte)

Zeigen Sie durch vollständige Induktion, dass $T(n) \leq 7n^2 - 6n$, falls n eine Zweierpotenz ist und $T(n)$ durch folgende Rekurrenzrelation beschrieben wird:

$$T(n) \leq \begin{cases} 1 & \text{falls } n = 1, \\ 4 \cdot T(\lceil \frac{n}{2} \rceil) + 6 \cdot n & \text{falls } n \geq 2. \end{cases}$$

A.2 Asymptotische Schranken finden und beweisen (2 Punkte)

Seien c_0 und c_1 positive Konstanten und

$$T(n) = \begin{cases} c_0 n & \text{falls } n \leq n_0, \\ T(\lfloor \frac{n}{5} \rfloor) + T(\lfloor \frac{7}{10} n + 2.5 \rfloor) + c_1 n & \text{falls } n > n_0. \end{cases}$$

für geeignet gewähltes $n_0 > 35$. Finden Sie eine Funktion f (in geschlossener Form), so dass $T(n) = \Theta(f(n))$ gilt und beweisen Sie ihre Behauptung.

B Master Theorem (2 Punkte)

Zeigen Sie asymptotische Schranken für die folgenden Rekurrenzen unter Anwendung des Mastertheorems.

$$T_1(n) = \begin{cases} 1 & \text{für } n = 1 \\ \sqrt{2} T_1(\frac{n}{2}) + c^2 n & \text{für } n = 2^k, k \in \mathbb{N} \end{cases} \tag{1}$$

$$T_2(n) = \begin{cases} 1 & \text{für } n = 1 \\ 4T_2(\frac{n}{4}) + 4n & \text{für } n = 4^k, k \in \mathbb{N} \end{cases} \tag{2}$$

$$T_3(n) = \begin{cases} 1 & \text{für } n = 1 \\ 8T_3(\frac{n}{2}) + n & \text{für } n = 2^k, k \in \mathbb{N} \end{cases} \tag{3}$$

$$T_4(n) = \begin{cases} 1 & \text{für } n = 1 \\ 3(T_4(\frac{n}{5}) + 2n + 1) + 3 & \text{für } n = 5^k, k \in \mathbb{N} \end{cases} \tag{4}$$

C Merge-Algorithmus (4 Punkte)

Der Merge-Algorithmus verschmilzt zwei sortierte Zahlenreihen zu einer sortierten Zahlenreihe. Gegeben sind also zwei aufsteigend sortierte Arrays $A[1, \ldots, n_1]$ und $B[1, \ldots n_2]$ von natürlichen Zahlen, und gesucht ist das aufsteigend sortierte Array $C[1, \ldots, (n_1 + n_2)]$ von Zahlen in A und B.

Geben Sie einen Algorithmus an, der das Merging-Problem in $\Theta(n)$ löst (mit $n = n_1 + n_2$), und beweisen Sie die Laufzeit Ihres Algorithmus. Beweisen Sie außerdem die Korrektheit Ihres Algorithmus mit Hilfe von Invarianten. (Tipp: Versuchen Sie eine Schleifeninvariante über eine sortierte Teilliste zu beweisen.)

Karlsruher Institut für Technologie

Algorithmen 1 – Sommersemester 2018
Zuverlässige Softwaresysteme im Kontext der Automobilindustrie
`http://verialg.iti.kit.edu`

Prof. Dr. Carsten Sinz – Markus Iser

Deckblatt Übungsblatt 2

Tutoriumsnummer:

Name	Matrikelnummer	Unterschrift

Mit unseren Unterschriften bestätigen wir, dass wir die Aufgaben eigenständig gelöst haben.

Hinweis: Das Übungsblatt darf in Gruppen von bis zu zwei Personen bearbeitet werden. Beide Personen müssen demselben Tutorium zugeteilt sein. Möchte jemand seine Abgaben-Gruppe innerhalb des Semesters wechseln, so ist dies im Voraus mit dem Tutor abzusprechen. **Bitte tragen Sie oben *groß* die Nummer Ihres Tutoriums ein.** Die Lösung des Übungsblattes ist in jedem Fall mit diesem Deckblatt abzugeben.

Bewertung (durch Tutor):