Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Dr. Hanjo Täubig
Jeremias Weihmann

Sommersemester 2012
Übungsblatt 7
4. Juni 2012

Grundlagen: Algorithmen und Datenstrukturen

Abgabetermin: Jeweilige Tutorübung in der Woche vom 11. bis 15. Juni

Tutoraufgabe 1

Wir haben uns in der Vorlesung beim Beweis, dass die Laufzeit von MergeSort in $\mathcal{O}(n \log n)$ liegt, auf das Mastertheorem berufen.
Zeigen Sie ohne Verwendung des Mastertheorems, dass die geschlossende Darstellung der Rekursionsformel

$$\begin{aligned} T(n) &= T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + T\left(\left\lceil \frac{n}{2} \right\rceil\right) + \Theta(n) && (1) \\ T(1) &= \Theta(1) && (2) \end{aligned}$$

für die Eingabemenge aller Zweierpotenzen in $\mathcal{O}(n \log n)$ liegt.

Tutoraufgabe 2

In der ersten Tutoraufgabe wurde gezeigt, dass die geschlossende Darstellung der Rekursionsformel

$$\begin{aligned} T(n) &= T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + T\left(\left\lceil \frac{n}{2} \right\rceil\right) + \Theta(n) \\ T(1) &= \Theta(1) \end{aligned}$$

für die Eingabemenge aller Zweierpotenzen in $\mathcal{O}(n \log n)$ liegt. Zeigen Sie, dass dies auch dann gilt, wenn alle natürlichen Zahlen als Eingabe erlaubt sind.

Tutoraufgabe 3

Ein Hotelmanager hat n Buchungen für die nächste Saison. Sein Hotel hat k identische Räume. Die Buchungen enthalten ein Ankunfts- und ein Abreisedatum. Er will herausfinden, ob er zu allen Zeiten genügend Räume für die Buchungen zur Verfügung hat. Entwickeln Sie einen Algorithmus, der dieses Problem in Zeit $\mathcal{O}(n + \text{sort}(n))$ löst, wobei $\text{sort}(n)$ die Worst-Case-Laufzeit eines beliebigen Algorithmus für das Sortieren von n Zahlen ist.

Hausaufgabe 1

Implementieren Sie in der Klasse `UIsmsArray` den MergeSort-Algorithmus, in der Funktion `sort`, der die Elemente in dem Feld `A` sortiert.
Verwenden Sie für Ihre Implementierung die auf der Übungswebseite bereitgestellten Klassen und verändern Sie für Ihre Implementierung *ausschließlich* die Klasse `UIsmsArray`.
Achten Sie bei der Abgabe Ihrer Aufgabe darauf, dass Ihre Klasse `UIsmsArray` heißt und auf den Rechnern der Rayhalle (`rayhalle.informatik.tu-muenchen.de`) mit der bereitgestellten Datei `main_m` kompiliert werden kann. Anderenfalls kann eine Korrektur nicht garantiert werden. Achten Sie darauf, dass Ihr Quelltext ausreichend kommentiert ist.
Schicken Sie die Lösung per Email mit dem Betreff `[GAD] Gruppe <Gruppennummer>` an Ihren Tutor.

Hausaufgabe 2

Wir betrachten die wie folgt definierte Funktion:

$$T(n) \leq \begin{cases} T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + T\left(\left\lceil \frac{n}{2} \right\rceil\right) + n, & \text{falls } n > 1 \\ 1, & \text{falls } n = 1 \end{cases} \tag{3}$$

In der ersten Tutoraufgabe wurde mit vollständiger Induktion gezeigt, dass

$$T(n) \leq n + n \operatorname{ld} n \tag{4}$$

für alle Zweierpotenzen n ist. Daraus folgt natürlich automatisch, dass

$$T(n) \leq n + n \operatorname{ld}(n) + 1/n \tag{5}$$

für alle Zweierpotenzen n gilt.

Zeigen Sie am Beispiel der Funktion T und den genannten oberen Schranken, dass es bei der Beweismethode über vollständige Induktion (analog wie in der ersten Tutoraufgabe durchgeführt) zu dem Phänomen kommen kann, dass der unmittelbare Beweis einer schwachen Schranke (5) fehlschlägt, während der unmittelbare Beweis einer starken Schranke (4) gelingt.

Hausaufgabe 3

Geben Sie ein Verfahren an, um 5 Elemente mit 7 Vergleichen zu sortieren.