
Grundlagen: Algorithmen und Datenstrukturen

Abgabetermin: Jeweilige Tutorübung in der Woche vom 10. bis 14. Juni

Tutoraufgabe 1

In dieser Aufgabe betrachten wir erneut die Laufzeitanalyse von MergeSort. In der Vorlesung wurde die Laufzeit rekursiv formuliert und das Mastertheorem verwendet, um zu zeigen, dass die Laufzeit von MergeSort in $\mathcal{O}(n \log n)$ liegt.

Die rekursive Formulierung der Laufzeit war wie folgt:

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

Zeigen Sie ohne Verwendung des Master-Theorems, dass die Worst-Case-Laufzeit von MergeSort angewandt auf Zahlenfolgen, deren Länge Zweierpotenzen sind, in $\mathcal{O}(n \log n)$ liegt.

- (a) Verwenden Sie dazu die Methode des iterativen Einsetzens.
- (b) Verwenden Sie dazu (starke) vollständige Induktion.

Tutoraufgabe 2

Wir betrachten eine Funktion $T(n)$, die für zwei Konstanten $a, c > 0$ die folgenden Eigenschaften besitzt:

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

In der ersten Tutoraufgabe wurde gezeigt, dass $T(n) \in \mathcal{O}(n \log n)$ für alle Zweierpotenzen n gilt. Zeigen Sie, dass dies auch dann gilt, wenn alle natürlichen Zahlen, und nicht nur Zweierpotenzen, als Eingabe erlaubt sind. Was folgt daraus für die Worst-Case-Laufzeit von MergeSort?

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 Linuxhalle (`lxhalle.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:

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

In der ersten Tutoraufgabe wurde mit (starker) vollständiger Induktion gezeigt, dass

$$T(n) \leq 2n \lg(2n) \quad (3)$$

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

$$T(n) \leq 2n \lg(2n) + 1/n \quad (4)$$

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 (4) fehlschlägt, während der unmittelbare Beweis einer starken Schranke (3) gelingt.

Hausaufgabe 3

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