

Übungsblatt 1

Ausgabe: 25.10.2022
Abgabe: 01.11.2022, **08:00**

Die Übungsgruppen wurden am 21.10. durch das [AUGE](#)-System zugeteilt. Sie können das Ergebnis dort abrufen. Außerdem wurden am 21.10. die Links für die Onlineabgabe verschickt. Sollten Sie bis zum 27.10. diese E-Mail nicht bekommen haben, aber am Übungsbetrieb teilnehmen wollen, schreiben Sie umgehend an algo222@cs.uni-frankfurt.de. Bitte beachten Sie, dass die Zusendung an Ihre HRZ-Mailadresse (...@stud.uni-frankfurt.de) erfolgt ist (überprüfen Sie auch Ihren Spam-Ordner). Abgaben per E-Mail sind *nicht vorgesehen*.

Durch erfolgreiche Bearbeitung der Übungsaufgaben können Sie einen Bonus von bis zu 10% für die Klausur erwerben, sofern Sie mindestens einmal vorgerechnet und die Klausur bestanden haben. Die Aufgaben sind grundsätzlich ohne zusätzliche Tipps und Hilfestellungen lösbar. Sollten zusätzliche Informationen nötig sein, wird es einen schriftlichen Hinweis auf dem entsprechenden Aufgabenblatt geben. **Antworten müssen stets begründet werden, sofern der Aufgabentext Sie nicht explizit davon befreit!** Weitere Informationen zum Übungsbetrieb finden Sie auf der [Webseite](#) zur Veranstaltung.

Aufgabe 1.1 *Einfache Sortieralgorithmen* (6 + 6 Punkte)

Bestimmen Sie für *Insertion Sort*, *Bubble Sort* und *Selection Sort* jeweils asymptotisch exakt (in Θ -Notation) die Anzahl der Vergleiche und die Anzahl der Vertauschungen in Abhängigkeit von N auf den folgenden Eingaben:

- a) 2, 1, 4, 3, 6, 5, ..., N , $N - 1$
- b) 1, $\frac{N}{2} + 1$, 2, $\frac{N}{2} + 2$, 3, $\frac{N}{2} + 3$, ..., $\frac{N}{2}$, N

Sie können davon ausgehen, dass N gerade ist.

Aufgabe 1.2 *Quicksort* (5 + 5 Punkte)

Sortieren Sie folgende Eingaben mit Quicksort und benutzen Sie dabei die jeweils angegebene Pivotwahl. Geben Sie die Reihenfolge der Zahlen jeweils vor den rekursiven Aufrufen der Prozedur `quicksort` an und markieren Sie sowohl das Pivotelement als auch die zwei zu sortierenden Teilarrays. Es ist nach dem Schema der `partition`-Funktion aus der Vorlesung vorzugehen.

- a) 77, 51, 64, 65, 17, 96, 82, 23 mit `pivot(links, rechts) = rechts`.
- b) 77, 51, 64, 65, 17, 96, 82, 23 mit `pivot(links, rechts) = links`.

Aufgabe 1.3 *Erwartete Laufzeit von Insertion Sort*

(2 + 3 + 3 Punkte)

In dieser Aufgabe soll eine untere Schranke für die erwartete Laufzeit von Insertion Sort hergeleitet werden, wenn alle Permutationen gleichwahrscheinlich sind.

- a) Betrachten Sie den Beginn des i -ten Durchlaufs der äußeren Schleife. Die Elemente $A[1], \dots, A[i]$ sind sortiert, und im nächsten Schritt wird das Element $A[i+1]$ in diese Folge einsortiert. Bestimmen Sie die Wahrscheinlichkeit für das folgende Ereignis in Abhängigkeit von i und j : $A[i+1]$ steht nach dem Einsortieren j Positionen weiter links.
- b) Bestimmen Sie mit dem Ergebnis aus a) den Erwartungswert der (zufälligen) Anzahl von Positionen, um die das Element $A[i+1]$ nach Einsortieren in $A[1], \dots, A[i]$ verschoben wurde in Abhängigkeit von i .
- c) Leiten Sie aus a) und b) folgende Aussage her:
Der Insertion Sort-Algorithmus hat eine erwartete Laufzeit von $\Omega(n^2)$.

Aufgabe 1.4 *Lastwagen (über-)laden*

(10 Punkte)

Ein Logistikunternehmen besitzt n Lastwagen, deren Nutzlasten L_1, \dots, L_n in einer Tabelle aufgelistet sind. Mit diesen Lastwagen sollen n Container transportiert werden, deren Gewichte G_1, \dots, G_n ebenfalls in einer Tabelle stehen. Jeder der Lastwagen kann genau einen Container transportieren. Befindet sich auf einem Lastwagen ein Container, dessen Gewicht die Nutzlast des Lastwagens übersteigt, ist der Lastwagen überladen.

Entwerfen Sie einen Algorithmus, der entscheidet, ob es möglich ist, die Container so auf die Lastwagen zu verteilen, dass kein Lastwagen überladen ist. Die asymptotische worst-case Laufzeit soll $o(n^2)$ sein. Zeigen Sie, dass die Laufzeitschranke eingehalten wird und der Algorithmus korrekt arbeitet.