

Grundlagen: Algorithmen und Datenstrukturen

Woche 4

Tobias Eppacher

School of Computation, Information and Technology

19. Mai 2025

Table of contents

Aufgaben

E-Aufgaben

Hausaufgaben

Aufgabe 4.1 - Sortierv Verfahren

Zahlenfolge: [523, 126, 67, 1, 500, 34, 21, 229, 9, 123, 13]

(a) Mergesort:

Ein Zwischenschritt je Aufteilen und Verschmelzen (hier ca. 8 Schritte).

Teilen ungerader Länge → Linke Seite eins mehr.

(b) Quicksort:

Wie in der Vorlesung → letztes Element = Pivot.

Machen Sie kenntlich welches Pivot in jedem Schritt verwendet wird und geben Sie das Array nach jedem Umsortieren an.

Sortierschritte auf zwei nicht überlagernden Abschnitten der Zahlenfolge können im selben Schritt dargestellt werden.

Bei Teilsequenzen der Länge 2 kann direkt ohne Rekursion getauscht werden.

(c) Welches Sortierv Verfahren ist im Worst- bzw. Averagecase schneller?

Aufgabe 4.1 - Sortiervverfahren (a)

523	126	67	1	500	34	21	229	9	123	13
-----	-----	----	---	-----	----	----	-----	---	-----	----

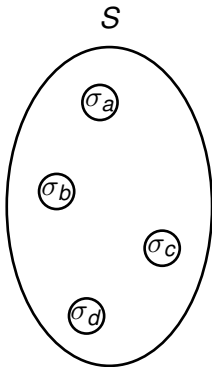
Aufgabe 4.1 - Sortiervverfahren (b)

523	126	67	1	500	34	21	229	9	123	13
-----	-----	----	---	-----	----	----	-----	---	-----	----

Aufgabe 4.1 - Sortiervverfahren (c)

Welches Sortiervverfahren ist im Worst- bzw. Averagecase schneller?

Aufgabe 4.2 - Bankkonto Tutorial



Menge von Operationen: S

(Obere Schranke für) Laufzeit von Operation
 $\sigma \in S: T(\sigma)$

(Obere Schranke für) Laufzeit einer Folge
von m Operationen:

$$T(\sigma_1, \sigma_2, \dots, \sigma_m) := \sum_{i=1}^m T(\sigma_i)$$

Ziel der Amortisierten Analyse

Möglichst genaue obere Schranke für
 $T(\sigma_1, \sigma_2, \dots, \sigma_m)$ finden

Aufgabe 4.2 - Bankkonto Tutorial

Bankkonto Methode

Idee: Verteile Laufzeit von langsamen auf schnellere Operationen

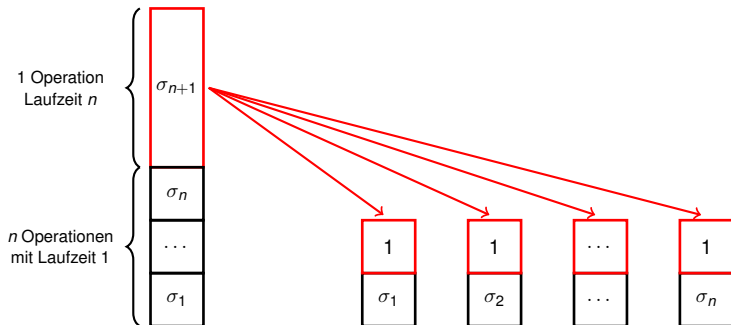
⇒ Bessere Abschätzung der Laufzeit

Aufgabe 4.2 - Bankkonto Tutorial

Bankkonto Methode

Idee: Verteile Laufzeit von langsamen auf schnellere Operationen

⇒ Bessere Abschätzung der Laufzeit



Aufgabe 4.2 - Bankkonto Tutorial

- ▶ Schnelle Operationen können Laufzeit übernehmen
⇒ Zahlen Tokens auf Konto ein ("nicht verwendete Zeit")
- ▶ Mehrere schnelle Operationen sparen Tokens an
- ▶ Folgende langsame Operation kann Tokens abheben
⇒ Gesparte Zeit wird aufgebraucht

Wir definieren die Funktion $\Delta : S \rightarrow \mathbb{R}$ die bestimmt, wieviele Tokens eine Operation einzahlt oder abhebt.

Diese muss zwei Eigenschaften erfüllen:

1. Für legale Operationsfolgen: $\sum_{i=1}^m \Delta(\sigma_i) \geq 0$
(Konto darf nie negativ werden)
2. Δ ist möglichst gut gewählt. (*Dazu gleich mehr*)

E-Aufgaben

- ▶ Aufgabe 3.5 - Noch mehr Spaß mit \mathcal{O}
 - ▶ Gute Überprüfung für Verständnis von \mathcal{O}
 - ▶ Kleiner Induktionsbeweis
- ▶ Aufgabe 3.6 - Zufallsvariablen-Caching
 - ▶ Übung zu Zufallsvariablen

Hausaufgaben

- ▶ Hausaufgabe 3 - Dynamisches Array
(Deadline: 21.05.2025)

Fragen?

- ▶ Nach Übung gerne bei mir melden
- ▶ Tutoriumschannel oder DM an mich auf Zulip
- ▶ Vorlesungschannels von GAD auf Zulip (insbesondere bei Hausaufgaben)

Feedback oder Verbesserungsvorschläge?

Gerne nach dem Tutorium mit mir quatschen oder DM auf Zulip

Bis nächste Woche!