Grundlagen: Algorithmen und Datenstrukturen

Woche 5

Tobias Eppacher

School of Computation, Information and Technology

26. Mai 2025

Inhalt

Aufgaben

E-Aufgaben

Hausaufgaben

Aufgabe 5.1 - Laufzeitanalyse: Mergesort

In dieser Aufgabe machen wir eine Laufzeitanalyse von Mergesort auf drei verschiedene Weisen.

a. **Argumentativ:** Wie viele Rekursionsebenen (ohne initialen Aufruf)? Asymptotischer Aufwand für jede Rekursionsebene? Was ist damit der asymptotische Aufwand für den gesamten Algorithmus?

b. **Iteratives Einsetzen**

c. **Vollständige Induktion**

Rekursive Formulierung der Laufzeit:

$$T(n) = T(\lfloor \frac{n}{2} \rfloor) + T(\lceil \frac{n}{2} \rceil) + \Theta(n)$$
$$T(1) = \Theta(1)$$

Aufgabe 5.1 - Laufzeitanalyse: Mergesort (a)

Aufgabe 5.1 - Laufzeitanalyse: Mergesort (b/c Anmerkungen)

Aufgabe 5.1 - Laufzeitanalyse: Mergesort (b)

Aufgabe 5.1 - Laufzeitanalyse: Mergesort (c)

Aufgabe 5.2 - Betrunkener Übungsleiter

Wir betrachten einen torkelnden Übungsleiter an einer Kletterwand, der verschiedene Operationen ausführen kann.

Starthöhe: $h = 0$

Operation	Höhenänderung	Laufzeit
`hoch`	$h = h + 1$	$T = 1$
`runter(int k)`	$h = h - \min\{h, k\}$	$T = \min\{h, k\}$

Anmerkung: `runter` *verringert die Höhe nie unter* 0*!*

Zeigen Sie mithilfe der Bankkonto-Methode, dass die amortisierten Laufzeiten der Operationen `hoch` und `runter(int k)` in $\mathcal{O}(1)$ liegen.

Aufgabe 5.2 - Betrunkener Übungsleiter

Aufgabe 5.2 - Betrunkener Übungsleiter

Aufgabe 5.3 - Stapelschlange

```
class Stapelschlange {
  private Stack s1 = new Stack();
  private Stack s2 = new Stack();

  public void enqueue(int v) {
          s1.push(v);
  }

  public int dequeue() {
    if(s2.isEmpty())
      while (!s1.isEmpty())
        s2.push(s1.pop());
    return s2.pop();
  }
}
```

Stack-Methode	Laufzeit
void push(int v)	$\mathcal{O}(1)$
int pop()	$\mathcal{O}(1)$
boolean isEmpty()	$\mathcal{O}(1)$

a. Was ist die Worst-Case Laufzeit von `dequeue` (abhängig von Queuegröße n)?

b. Finden Sie ein Amortisierungsschema, das die amortisierte Laufzeit der Operationen minimiert.
Geben Sie die Laufzeiten an und zeigen Sie die Korrektheit des Schemas?

Aufgabe 5.3 - Stapelschlange (a)

Aufgabe 5.3 - Stapelschlange (b)

Aufgabe 5.3 - Stapelschlange (b)

E-Aufgaben

- Aufgabe 5.4 - Instabile Sortierverfahren
  - Worstcase Suche für verschiedene Pivot-Wahlen
- Aufgabe 5.5 - Datenstruktur Amore
  - Amortisierte Analyse

Hausaufgaben

- Hausaufgabe 3 - Dynamisches Array
  (Deadline: 28.05.2025)
- Hausaufgabe 4 - Verbessertes Sortieren
  (Deadline: 28.05.2025)
- Hausaufgabe 5 - Radixsort
  (Deadline: 04.06.2025)

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!