

## DAP2 – Präsenzübung 8

Besprechung: 06.06.2018 — 08.06.2018

### Abgabe:

Präsenzübungen müssen nicht zu Hause bearbeitet werden, sondern werden unter Anleitung während der Übung erarbeitet.

### Präsenzaufgabe 8.1: (AVL-Bäume)

**Hinweis:** In dieser Aufgabe dürfen die Knoten der Bäume ohne ihre Werte gezeichnet werden.

1. Zeichnen Sie einen AVL-Baum der Höhe 4 mit einer minimalen Anzahl an Knoten (ein Baum mit einem Knoten hat die Höhe Null).
2. Zeichnen Sie zwei AVL-Bäume mit je 12 Knoten, einmal mit einer minimalen Anzahl an Blättern und einmal mit einer maximalen Anzahl an Blättern.
3. In einen leeren AVL-Baum sollen die Zahlen **1, 2, 3, 4, 5, 6, 7, 8, 9** und **10** eingefügt werden. Geben Sie eine Einfüge-Reihenfolge an, bei der keine Rotationen im AVL-Baum nötig sind.
4. Formulieren Sie einen Algorithmus als Pseudocode, der rekursiv alle Zahlen aus einem gegebenen AVL-Baum ausgibt, die in dem Intervall  $[a, b]$  liegen. Dabei soll eine minimale Anzahl an Knoten besucht werden. Die Reihenfolge der ausgegeben Zahlen soll aufsteigend erfolgen (ohne dass die Zahlen noch extra sortiert werden).

### Präsenzaufgabe 8.2: (Datenstrukturen)

Entwerfen Sie eine Datenstruktur, die alle folgenden Operationen in  $\mathcal{O}(1)$  unterstützt. In dieser Aufgabe sind alle Elemente aus dem Wertebereich  $\{1, \dots, k\}$ , **und  $k$  ist eine Konstante.**

- **Einfügen( $x$ ):** Fügt ein Element  $x$  in die Datenstruktur ein.
- **LöscheNeuestes:** Löscht unter allen Elementen in der Datenstruktur das, was zuletzt eingefügt wurde.
- **LöscheÄltestes:** Löscht unter allen Elementen in der Datenstruktur das, was am frühesten eingefügt wurde.
- **Zähle( $x$ ):** Gibt zurück, wie oft  $x$  aktuell in der Datenstruktur ist.

Beispiel: Nach der Sequenz

Einfügen(8), Einfügen(8), Einfügen(3), Einfügen(8), Einfügen(5), LöscheÄltestes, Zähle(8)

soll die Datenstruktur bei der letzten Operation 2 zurückliefern, da die 8 noch zweimal in der Datenstruktur ist.

Für die volle Punktzahl wird erwartet, dass **alle Operationen eine Worst-Case Laufzeit von  $\mathcal{O}(1)$  haben.**

Beschreiben Sie in wenigen kurzen Sätzen, wie Ihre Datenstruktur aufgebaut ist und wie die angegebenen Operationen realisiert werden. Hierbei ist kein Pseudocode gefordert. Machen Sie deutlich, dass die Datenstruktur korrekt arbeitet und alle Operationen Worst-Case-Laufzeit  $\mathcal{O}(1)$  haben.