

# Algorithmen und Datenstrukturen

## Sommersemester 2022

### Woche 8

Kevin Angele, Tobias Dick, Oskar Neuhuber,  
Andrea Portscher, Monika Steidl, Laurin Wischounig

Abgabe bis 17.05.2022 23:59  
Besprechung im PS am 19.05.2022

#### Aufgabe 1 (2 Punkte): Kollisionensbehandlung

Kollisionen sind unvermeidbar bei Hashing - deshalb gibt es verschiedene Methoden damit umzugehen. Gegeben ist folgende Hash-Funktion:

$$f1(k) = (3k + 3) \% 11 \quad (1)$$

Zeichnen Sie das Array nach jedem Einfügen auf. Folgende Schlüssel werden eingefügt:  
13-18-44-23-2-47-22-11.

- Verwenden Sie zur Kollisionsbehandlung lineare Sondierung.
- Verwenden Sie zur Kollisionsbehandlung externe Verkettung.
- Verwenden Sie zur Kollisionsbehandlung Doppeltes Hashing. Wählen Sie dafür eine passende Hash-Funktion.

#### Aufgabe 2 (3 Punkte): Mengenoperationen

Maps können verwendet werden, um Mengen zu modellieren, indem wir bei einem Key-Value-Paar nur den Key betrachten. In Java gibt es vorimplementierte Klassen für Mengen wie zB. HashSet. Implementieren Sie folgende übliche Operationen für Mengen: Schnittmenge (intersection), Vereinigungsmenge (union), Differenz (difference) und symmetrische Differenz (symmetricDifference). Vervollständigen sie dazu den vorgegebenen Code in `Sets.java`. Sie dürfen keine bereits implementierten Funktionen für diese Operationen (zB. `retainAll()`) verwenden.

#### Aufgabe 3 (3 Punkte): Datentyp, Datenstruktur & Algorithmus

Sie wollen einen möglichst effizienten Suchfilter implementieren, der Buchtitel in Ihrer Bibliothek verwaltet. Sie Suche soll wie folgt funktionieren:

Der/Die Nutzer/in tippt die ersten  $b$  Buchstaben des Buchtitels ein. Zurückgegeben wird die Anzahl  $n$  der Buchtitel, die genau mit der eingegebenen Buchstabensequenz beginnen.  $k$  ist eine voreingestellte Konstante die angibt wie viele Buchtitel ausgegeben werden sollen. Das bedeutet ist  $n \leq k$ , wird eine alphabetisch sortierte Liste der  $n$  Buchtitel ausgegeben.

1. Welchen abstrakten Datentyp und welche Datenstruktur würden Sie für diesen Suchfilter von Buchtitel verwenden, wenn sie die Abfrage möglichst effizient gestalten wollen? Begründen Sie Ihre Antwort.
2. Ihr gewählter abstrakter Datentyp ist auf Basis eines Arrays implementiert. Ist hier ein sortiertes oder unsortiertes Array besser geeignet?
3. Welche asymptotische Laufzeitkomplexität (als Groß-O-Funktion von  $k$ ,  $n$  und der Gesamtanzahl  $N$  der gespeicherten Buchtitel) beansprucht Ihr Algorithmus, um festzustellen, ob  $n \leq k$  ist oder nicht? Begründen Sie Ihre Antwort.
4. Ist  $n \leq k$ , welche asymptotische Laufzeitkomplexität (als Funktion von  $k$ ,  $n$  und  $N$ ) beansprucht Ihr Algorithmus, um die Liste der  $n$  Buchtitel zu erstellen? Begründen Sie Ihre Antwort.

#### Aufgabe 4 (2 Punkte): AVL-Baum

1. Fügen Sie die angegebenen Elemente in einem leeren AVL Baum ein. Zeichnen Sie den Baum nach jeder Operation auf und balancieren Sie ihn, falls notwendig.  
15-7-6-18-18-16-1-3-21-30-32
2. Was ist die Höhe des resultierenden Baumes?
3. Führen Sie eine Inorder-Traversierung durch - was fällt Ihnen bei der Ausgabe der Zahlen auf?
4. Löschen Sie die folgenden Elemente aus dem erzeugten AVL Baum. Zeichnen Sie den Baum nach jeder Operation auf und balancieren Sie ihn falls notwendig.  
18-15-1-3-30