

# Algorithmen und Datenstrukturen

## Sommersemester 2022

### Blatt 6

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

Abgabe bis 03.05.2022 23:59  
Besprechung im PS am 05.05.2022

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

In dieser Aufgabe sind Sie dafür zuständig ein System zu implementieren, welches ankommende Flugzeuge ihrer Priorität nach sortiert. Die Flugzeuge landen anschließend in der resultierenden Reihenfolge. Die Priorität eines ankommenden Flugzeuges setzt sich zusammen aus der verbleibenden Menge an Kerosin und der aktuellen Verspätung (kann auch negativ sein für früher ankommende Flugzeuge). Für die Sortierung der Flugzeuge müssen Sie folgende Bedingungen beachten.

- Höchste Priorität haben Flugzeuge mit weniger als 25 Prozent verbleibender Menge an Kerosin
- Falls es mehrere Flugzeuge mit weniger als 25 Prozent Kerosin gibt, wird das mit der geringsten Menge bevorzugt
- Falls alle ankommenden Flugzeuge genug Kerosin haben, so werden Flugzeuge mit höherer Verspätung bevorzugt
- Sollten alle ankommenden Flugzeuge genug Kerosin haben und keine Verspätung (bzw. früher als geplant ankommen; negative Verspätung), dann werden die Flugzeuge mit geringerer Menge an Kerosin bevorzugt

Schauen Sie sich für diese Aufgabe den Code in der Datei `FlightManagement.java` an. In der Datei befinden sich drei Klassen `FlightManagement`, `PlanePriorityComparator` und `IncomingPlane`. Ihre Aufgabe ist es, die Methode `compare` in der Klasse `PriorityComparator` zu implementieren. Führen Sie anschließend die `main`-Methode in der Klasse `FlightManagement` aus und überprüfen Sie, dass Ihre Ausgabe der erwarteten Ausgabe entspricht.

#### Aufgabe 2 (3 Punkte): Heap Construction

Gegeben ist das folgende Array an Schlüsseln: 

|    |    |    |    |    |    |    |   |
|----|----|----|----|----|----|----|---|
| 39 | 25 | 62 | 56 | 72 | 45 | 12 | 5 |
|----|----|----|----|----|----|----|---|

In der Vorlesung haben Sie die folgenden zwei Methoden zum Aufbau eines Heaps kennengelernt:

1. Einfügen jedes Schlüssels in einen zu Beginn leeren Heap

## 2. Bottom-Up Heap Construction

Bauen Sie für beide Methoden den Heap auf und geben Sie für alle Zwischenschritte sowohl die grafische Darstellung des Baums, als auch das Array zum Baum (d.h. den Baum in zeilenweiser Nummerierung) an.

## Aufgabe 3 (3 Punkte): Sortieren mit einer Vorrangwarteschlange

Eingabe: 

|    |    |    |    |    |    |    |   |
|----|----|----|----|----|----|----|---|
| 72 | 63 | 55 | 63 | 35 | 96 | 40 | 1 |
|----|----|----|----|----|----|----|---|

Sortieren Sie die gegebene Eingabe mittels der folgenden Sortieralgorithmen:

1. Heap Sort
2. Insertion Sort
3. Selection Sort

Geben Sie dazu die einzelnen Teilschritte (Arrays und beim Heap Sort den Baum) an und markieren Sie den sortierten Bereich. Welche der drei Sortieralgorithmen sind *stabil*?

## Aufgabe 4 (2 Punkte): Lastenverteilung

In dieser Aufgabe sollen Sie sich konzeptuell überlegen welche ADTen am besten geeignet sind um ankommende Netzwerkanfragen unterschiedlicher Priorität zu verwalten. Diese Anfragen sollen ihrer Priorität nach an zur Verfügung stehende Server verteilt werden. Der Server zur Verarbeitung einer Anfrage wird nach dem Round Robin Verfahren ausgewählt. Dies bedeutet, dass reihum immer der nächste freie Server verwendet wird. Sobald ein Server eine Anfrage abgearbeitet hat, steht dieser wieder für neue Anfragen zur Verfügung.

1. Welcher ADT eignet sich besten für die Verwaltung der ...
  - (a) ... ankommenden Anfragen? (Kurz begründen)
  - (b) ... zur Verfügung stehenden Server? (Kurz begründen)
2. Welche Implementierung des ADT würden Sie wählen wenn ...
  - (a) ... das Registrieren (hinzufügen zum ADT) von Anfragen möglichst effizient sein soll? (Kurz begründen)
  - (b) ... das Zuweisen von Anfragen (entnehmen aus dem ADT) zu Servern möglichst effizient sein soll? (Kurz begründen)