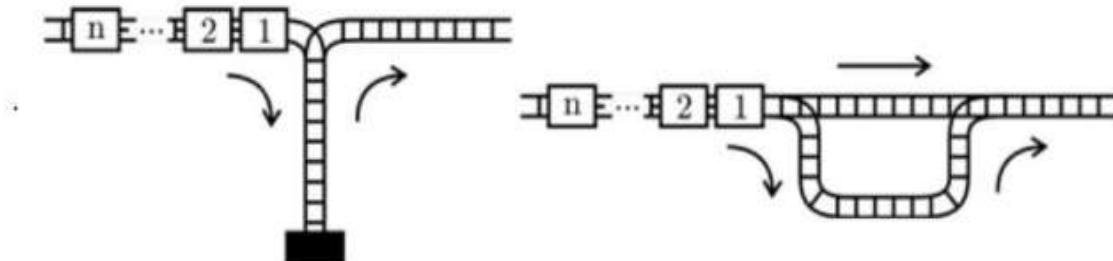


# Aufgaben zu Algorithmik & Datenstrukturen

## A1 – Stack und Queues

Die Abbildungen zeigen Eisenbahngleise, welche einen Stack bzw. eine Queue mit Überholspur darstellen.



Auf der linken Seite stehen die absteigend nummerierten Waggons  $n$  bis 1, die auf dem „Rangierwerk“ so umgestellt werden müssen, dass sie rechts in einer neuen Zusammenstellung herausfahren können.

Gibt es für einen Zug mit  $n = 5$  Waggons Rangiermöglichkeiten, so dass die folgenden Waggon-Zusammenstellungen auf der rechten Seite entstehen?

- a) 5, 4, 3, 2, 1                      c) 4, 2, 5, 3, 1  
 b) 5, 3, 1, 2, 4                      d) 1, 4, 5, 3, 2

## A2 – Sortieren

- a) Sortieren Sie das folgende Array durch Anwendung des Selectionsort-Algorithmus. Geben Sie dazu das Array nach jeder Swap-Operation an.

[7, 3, 6, 2, 1]

- b) Sortieren Sie das folgende Array durch Anwendung des Insertionsort-Algorithmus. Geben Sie dazu das Array nach jeder Iteration der äußersten Schleife an.

[7, 2, 3, 6, 1]

- c) Sortieren Sie das folgende Array durch Anwendung des Mergesort-Algorithmus. Geben Sie dazu das Eingabe-Array nach jeder Merge-Operation an.

[4, 7, 2, 9, 1, 8]

- d) Sortieren Sie das folgende Array durch Anwendung des Bubblesort-Algorithmus. Geben Sie dazu das Array nach jeder Swap-Operation an.

[6, 2, 3, 7, 1]

### A3 – Best- und Worstcase

Geben Sie für ein beliebiges, aber festes  $n \in \mathbb{N}$  ein Array der Länge  $n$  an, sodass der **Bubblesort**-Algorithmus so *vielen* **Swap**-Operationen wie möglich benötigt (Worst-Case) und zusätzlich ein Array der Länge  $n$ , sodass der Bubblesort-Algorithmus so *wenig* Swap-Operationen wie möglich benötigt (Best-Case). Begründen Sie Ihre Antworten kurz.

### A4 – Stack

In einen Stack werden die Zahlen von 0-9 in dieser Reihenfolge eingefügt.

#### Aufgabe:

Geben Sie, wenn möglich, die richtige Reihenfolge an, in der push- und pop-Befehle ausgeführt werden müssen, um die angegebene Zahlenfolge zu erreichen.

#### Beispiel:

- Aufgabe: 3 2 1 0 9 8 7 6 5 4
- Lösung: + + + + 3 2 1 0 + + + + + 9 8 7 6 5 4

$+$  = *push*, *Zahl* = *pop*

- a) 1 2 3 4 5 6 9 8 7 0
- b) 4 3 7 6 5 8 2 1 9 0
- c) 0 1 4 3 5 2 6 9 7 8
- d) 0 1 3 6 9 8 7 5 4 2