

Aufgabe: Rangierwerk (Stacks & Queues)

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

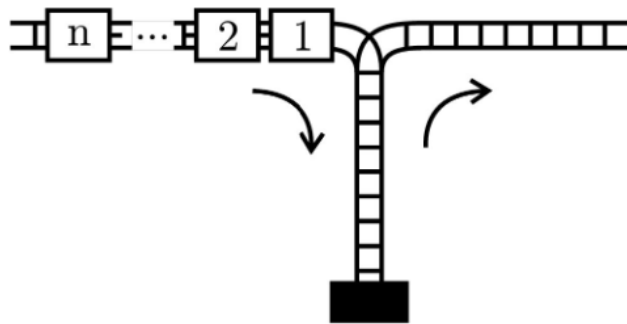


Abbildung 1: Stack

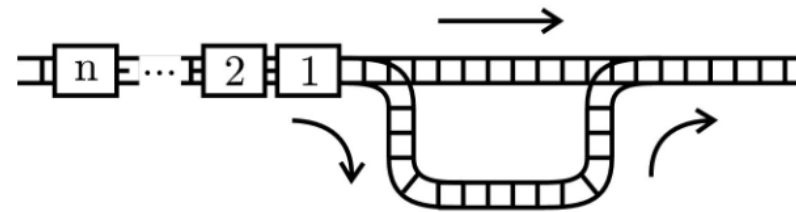


Abbildung 2: Queue

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

Aufgabe: ABC (Stacks & Queues)

Entwerfen Sie einen Algorithmus **ABC**(a, b, c, w), der bei Eingabe von Zeichen a, b, c und Wort $w = a^k b^l c^m$ ($k, l, m \geq 0$) testet, ob w die Form $a^n b^n c^n$ für ein $n \geq 0$ hat.

Die Zeichen a, b, c sind vom Datentyp Character. Das Wort w ist als Stack gegeben. Werte des Stacks sind vom Datentyp Character. Das 1. Zeichen des Wortes liegt oben auf dem Stack. Der Stack erlaubt die Operationen:

`push(value)`, `pop()`, `top()`, `isEmpty()`.

Als Datenstrukturen in Ihrem Algorithmus darf der gegebene Stack benutzt werden und (maximal) eine Queue. Die Queue erlaubt die Operationen:

`enqueue(element)`, `dequeue()`, `head()`, `isEmpty()`.

Ansonsten dürfen keine weiteren Datenstrukturen benutzt werden, auch keine primitiven Datentypen. Ausnahme: Es dürfen Boolesche Werte und Characters benutzt werden.