

# Übungen aus Algorithmen und Datenstrukturen

## Übung 6

### Backtracking und Pfadfindung

# Aufgabe 1

## Darstellung des Schachbretts

Ein  $n \times n$  Felder großes Schachbrett, auf dem Damen platziert werden sollen, kann als boole'sches Array dargestellt werden. Felder (d.h. Elemente des Arrays) auf denen Damen platziert sind werden mit `true`, Felder auf denen keine Dame steht mit `false` markiert.

## Prüfen einer neuen Damenposition auf Gültigkeit

Implementieren Sie einen Algorithmus

`boolean` checkPosition(`boolean`[][] array, `int` column, `int` line) welche prüft, ob für eine bestimmte Damenkonfiguration auf einem Schachbrett, beschrieben durch dass Array array, eine weitere Dame in der Spalte column und Zeile line gültig hinzugefügt werden kann.

## Aufgabe 2

### Hinzufügen einer weiteren Dame in einer bestimmten Zeile

Implementieren Sie einen Algorithmus `boolean placeQueen(boolean[][] array, int line)`, welcher rekursiv versucht eine Dame in der Zeile `line` gültig auf dem Schachbrett zu platzieren.

# Aufgabe 3

## Weg zwischen zwei Vertices in einem Graphen

Implementieren Sie einen Algorithmus

`boolean` existPath(`int`[][] adjazenzMatrix, `int` start, `int` end), welcher ermittelt, ob es in dem Graphen, beschrieben durch die Adjazenzmatrix adjazenzMatrix, zumindest einen Pfad zwischen den beiden Vertices start und end gibt.

## Hinweis

Auch dieser Algorithmus lässt sich als Backtracking-Algorithmus formulieren und er ist eine Variante der Tiefensuche.