

☐ Gruppe M. Hava☒ Gruppe J. HeinzelreiterName: Neuhold Michael

Aufwand [h]: \_\_\_\_\_

☐ Gruppe P. Kulczycki

Beispiel	Lösungsidee (max. 100%)	Implement. (max. 100%)	Testen (max. 100%)
1 (40 P)			
2 (60 P)			

**Beispiel 1: Pipeline (src/pipe/)**

Ein Installateur muss eine Strecke von  $x$  Metern mit Rohren überbrücken. Er hat ein Lager mit verschiedenen langen Rohren zur Verfügung und möchte wissen, ob er diese  $x$  Meter erreichen kann, ohne eines der lagernden Rohre zersägen zu müssen. Implementieren Sie für den Installateur eine Funktion

```
bool possible (int const x, int const lengths [], int const counts [], int const n);
```

die dem Installateur diese Frage beantwortet. Dabei seien `lengths` und `counts` zwei Felder mit jeweils  $n$  Elementen, die für die jeweiligen Rohrlängen samt den entsprechend verfügbaren Stückzahlen stehen.

Implementieren Sie `possible` in drei Varianten. Einmal als iterative Funktion (mit  $n$  geschachtelten `for`-Schleifen), einmal als rekursive Funktion und einmal als Funktion, die nach dem Backtracking-Schema arbeitet.

Testen Sie Ihre Funktionen ausführlich und geben Sie auch die (empirisch ermittelten) Laufzeiten für verschiedene Problemgrößen an.

**Beispiel 2: Sudoku (src/sudoku/)**

Implementieren Sie eine Funktion

```
void sudoku_solve (int squares []);
```

die ein in `squares` (der Größe  $9 \cdot 9 = 81$ ) gegebenes Sudoku nach dem Backtracking-Schema löst. Entnehmen Sie der Seite <https://en.wikipedia.org/wiki/Sudoku> alles Wesentliche über den Aufbau von Sudokus.

Testen Sie Ihre Funktion ausführlich und geben Sie auch die (empirisch ermittelten) Laufzeiten für Sudokus mit verschiedenen Schwierigkeitsgraden an.