

# Übungsserie 2: Lösungen

## Aufgabe 1: Laufzeit-Analyse

Es soll die Laufzeit des folgenden Programmfragments analysiert werden:

```
for i ← 0 to n-1 do
  for j ← i to n do
    S
```

mit  $T_s(n) \in O(1)$ .

Die Laufzeit soll in *expliziter* Form (als Polynom) und in der *O-Notation* angegeben werden.

Die allgemeine Summenformel (funktioniert auch für Folgen mit einer Anzahl von Summanden ungleich  $n$ ):

$$\sum a_i = a_1 m + \frac{m(m-1)}{2} d$$

wobei:  $m$  = Anzahl Summanden in der Summe  
 $d$  = Abstand zwischen den Summanden

### Lösung:

Die innere Schleife wird  $n+1, n, n-1, \dots, 3, 2$  mal ausgeführt.  
 Die Anzahl der Summanden ist somit:  $m = n$

Die Summenformel lautet also:

$$\begin{aligned} \sum_{i=2}^{n+1} a_i &= a_1 m + \frac{m(m-1)}{2} d \quad (m=n) \\ &= a_1 n + \frac{n(n-1)}{2} d \quad (a_1=2, d=1) \\ &= 2n + \frac{n(n-1)}{2} \\ &= \frac{4n + n^2 - n}{2} \\ &= \frac{n^2 + 3n}{2} \\ &\in O(n^2) \end{aligned}$$

## Aufgabe 2: Deque

Die bereits teilweise bestehende Implementierung einer *Double-Ended-Queue* (*Deque*/*Deque*) mit einer doppelt verketteten Liste soll vervollständigt werden.

### Lösung:

siehe: [uebung02/ml/aufgabe02/DequeImplementation.java](#)  
resp.: [uebung02/ml/aufgabe02/deque\\_implementation.py](#)

## Aufgabe 3: Stack

Es soll ein Stack mit einer einfach verketteten Liste implementiert werden. Dabei sollen alle Stack-Operationen mit einer Laufzeit von  $O(1)$  realisiert werden (mit Ausnahme der `print()`-Methode).

### Lösung:

siehe: [uebung02/ml/aufgabe03/StackImpl.java](#)  
resp.: [uebung02/ml/aufgabe03/stack\\_implementation.py](#)