

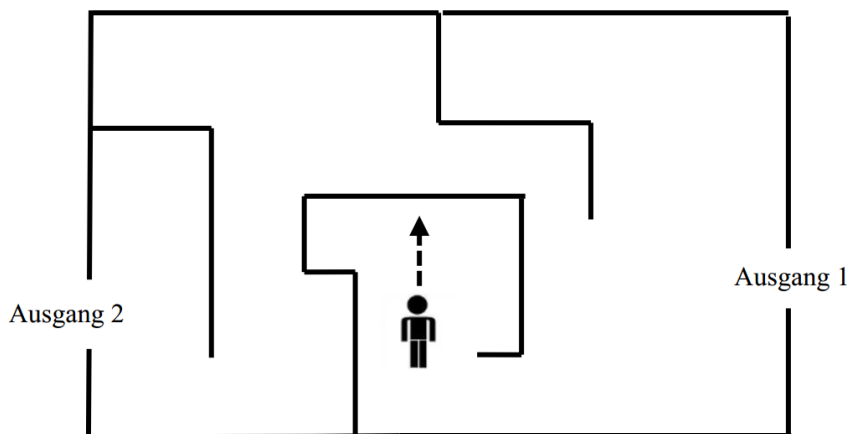


Übung 01: Pledge, Bubblesort, Landau-Symbole

Aufgabe 1: Pledge Algorithmus

Wenden Sie den Pledge Algorithmus¹ auf die folgende Abbildung an! Tragen Sie den zurückgelegten Weg, beschriftet mit den jeweiligen Werten des „Umdrehungszählers“, für die folgenden 2 Fälle ein:

- Es gibt nur Ausgang 1, Ausgang 2 ist verschlossen.
- Es gibt nur Ausgang 2, Ausgang 1 ist verschlossen.



Aufgabe 2: BubbleSort

Gegeben ist die folgende Implementierung des Sortieralgorithmus `Bubblesort`.

```
BUBBLE-SORT(A)
1 for  $i = 1$  to  $A.length - 1$ 
2   for  $j = A.length - 1$  downto  $i$ 
3     if  $A[j] < A[j-1]$ 
4       exchange  $A[j]$  with  $A[j-1]$ 
```

- Führen Sie den Algorithmus **manuell** für das folgende Eingabe-Array A aus und geben Sie die Belegung des Arrays **nach jeder Vertauschung** an:
 $A = \langle 35, 22, 10, 51, 48 \rangle$
- Invariante*: Was gilt jeweils am Ende einer Iteration der äußeren Vorschleife?
- Führen Sie den Algorithmus `InsertionSort` der Vorlesung **manuell** auf der Eingabe von Aufgabe b) aus. Geben Sie die jeweils die Belegung **am Ende der Iteration** der for-Schleife an.
- Was ist die asymptotische Laufzeit von `Bubblesort` im Worst Case! Überlegen Sie sich, für welche Eingabe der Worst Case eintritt und zählen Sie **die exakte Anzahl der Vertauschungen** (Zeile 4 im Code)! Geben Sie die Laufzeit am Schluss in der Θ -Notation an!
- Vergleichen Sie die **asymptotische Worst Case** Laufzeit von `Bubblesort` und `Insertionsort`!

(umblättern)

¹ Finale Version der Vorlesung, Startrichtung eingezeichnet.

- f) Implementieren Sie `BubbleSort` gemäß dem Pseudocode in Java. Sie können die vorgegebene Datei `BubbleSort.java` verwenden. Testen Sie auf Korrektheit! Der vorgegebene Code erzeugt ein absteigend sortiertes Array mit 4000 Integer zum Testen.
- g) Messen Sie in Ihrem Java-Programm die Laufzeit mit der Funktion `System.currentTimeMillis()`. Die Idee ist, dass Sie einen „Zeitstempel“ vor und nach dem Sortieren mitschneiden.
- Wie ändert sich die gemessene Zeit, wenn Sie das Eingabearray verdoppeln?
 - Warum ist diese Messung nicht aussagekräftig für die Laufzeit des Algorithmus?

Aufgabe 3: Asymptotisches Wachstum von Funktionen

Gelten die folgenden Aussagen? Begründen Sie jeweils Ihre Antwort!

Tipp: Finden Sie geeignete Konstanten!

- a) $2^{n+1} = O(2^n)$
- b) $2^{2n} = O(2^n)$
- c) Geben Sie für jedes Paar (A, B) an, ob die durch A beschriebene Funktion in $O(B)$, $\Omega(B)$ und $\Theta(B)$ liegt. Es genügt, wenn Sie einfach ja/nein eintragen.
Beispiel: Gilt, dass $n + 1 \in O(n)$?

A	B	O	Ω	Θ
$n + 1$	n			
$1 + \frac{1}{n}$	n^2			
$2n^3 - 15n^2 + n$	$4n^{3,5}$			
$4 \log_2 n$	$\log_{10} n$			