

Einführung Algorithmus

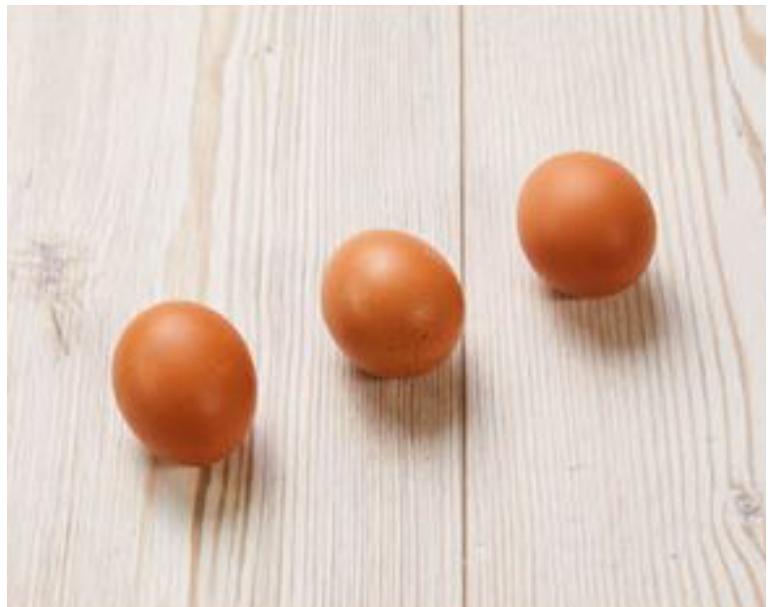
Was ist ein Algorithmus?

Ein Algorithmus ist eine *präzise, endliche* Abfolge von Anweisungen zur Lösung eines Problems.



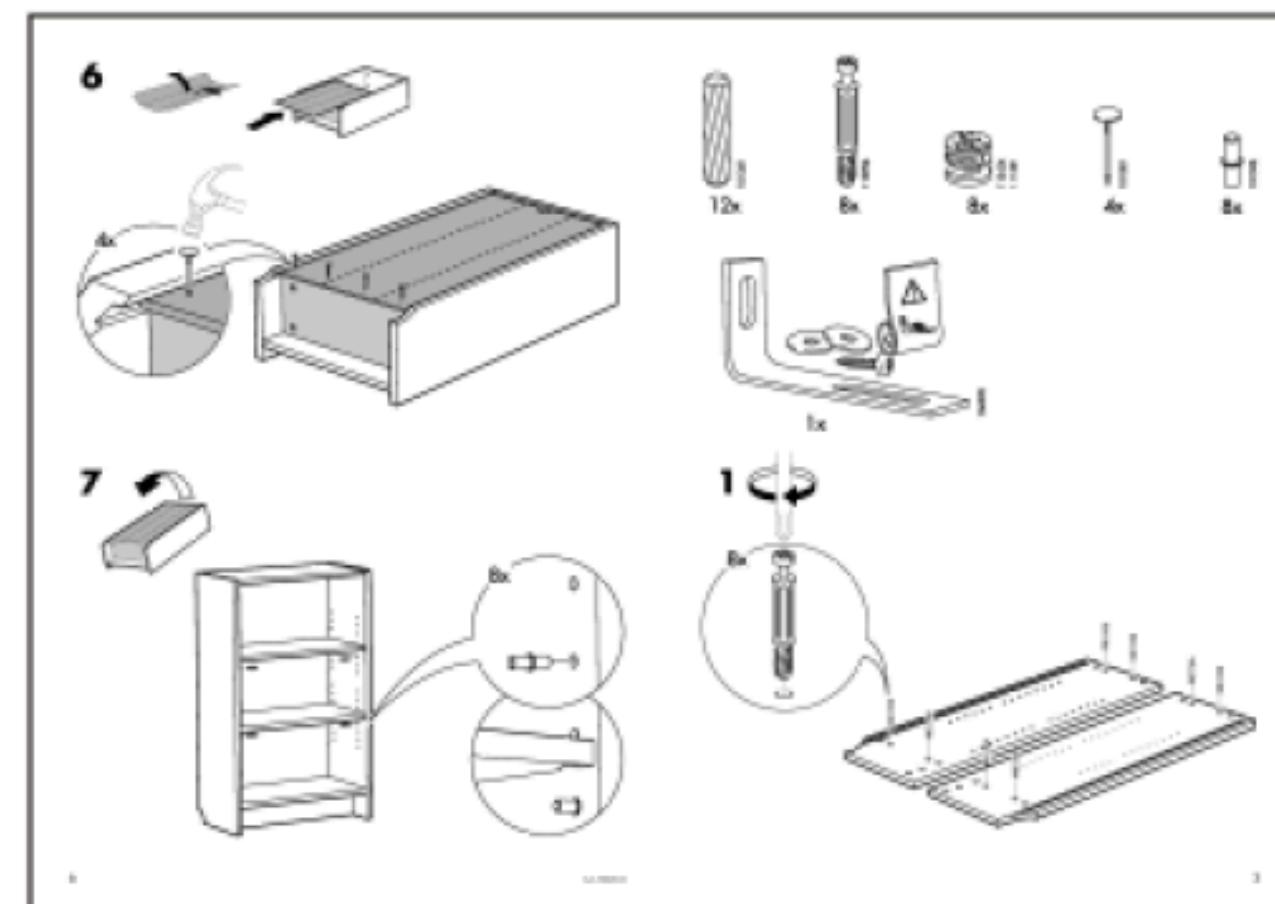
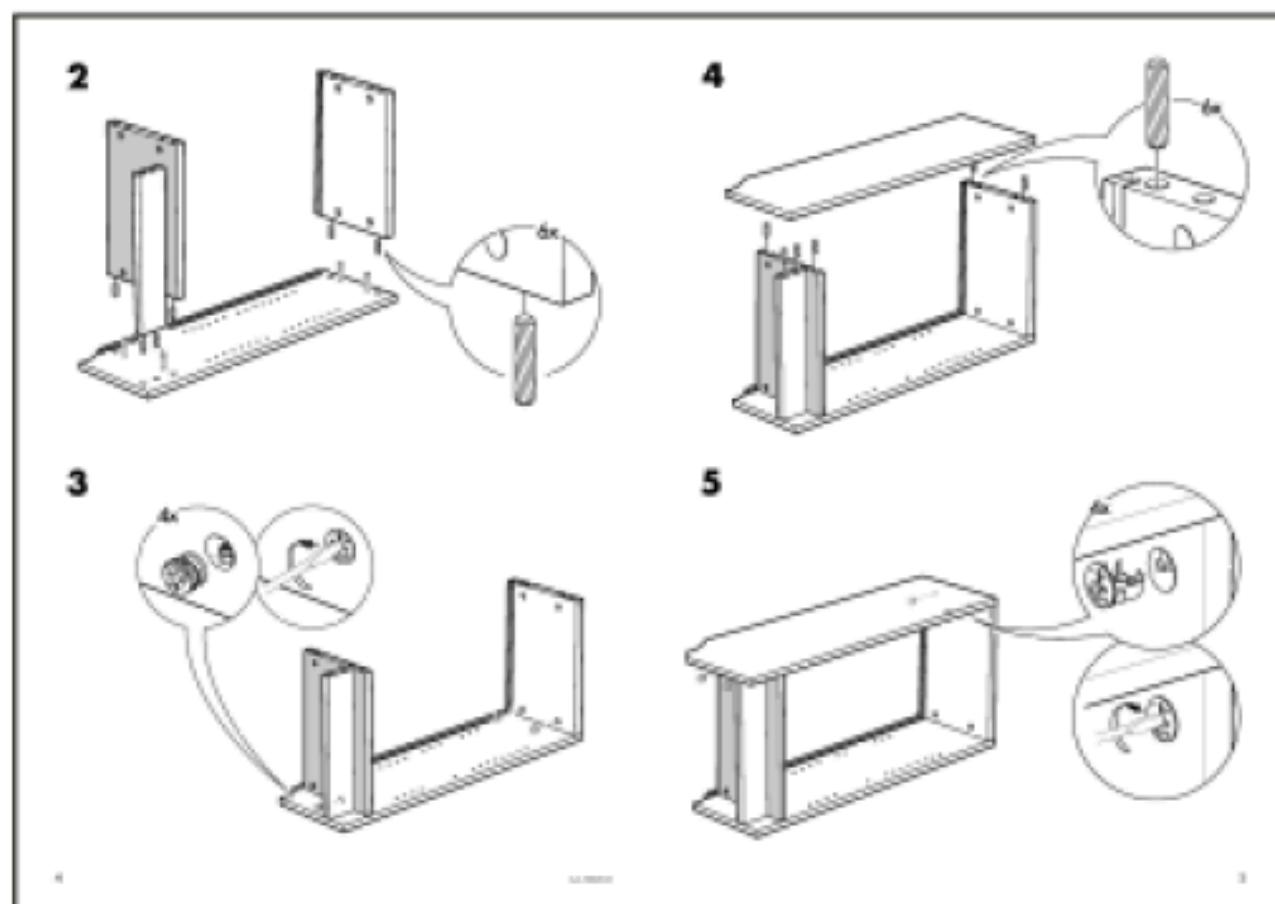
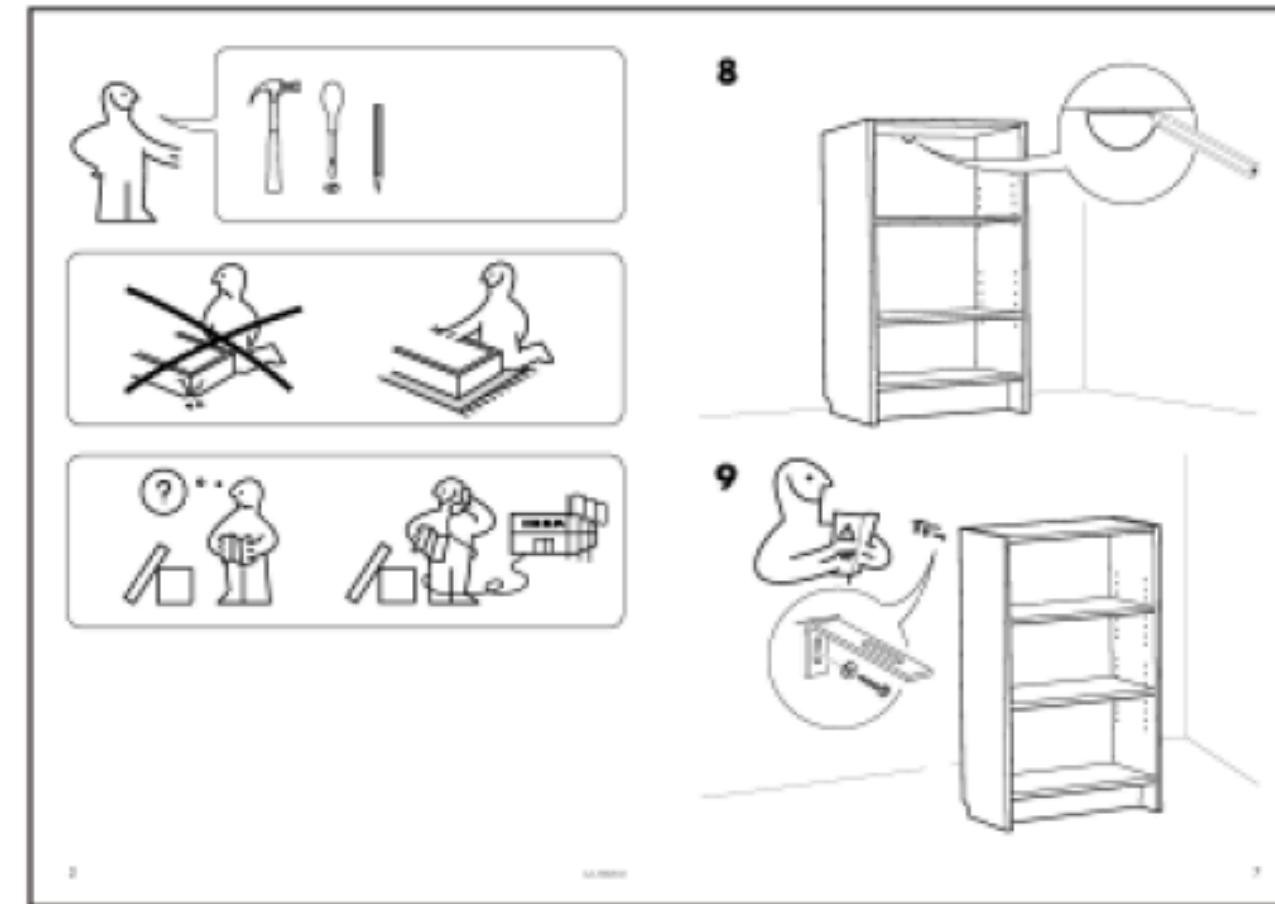
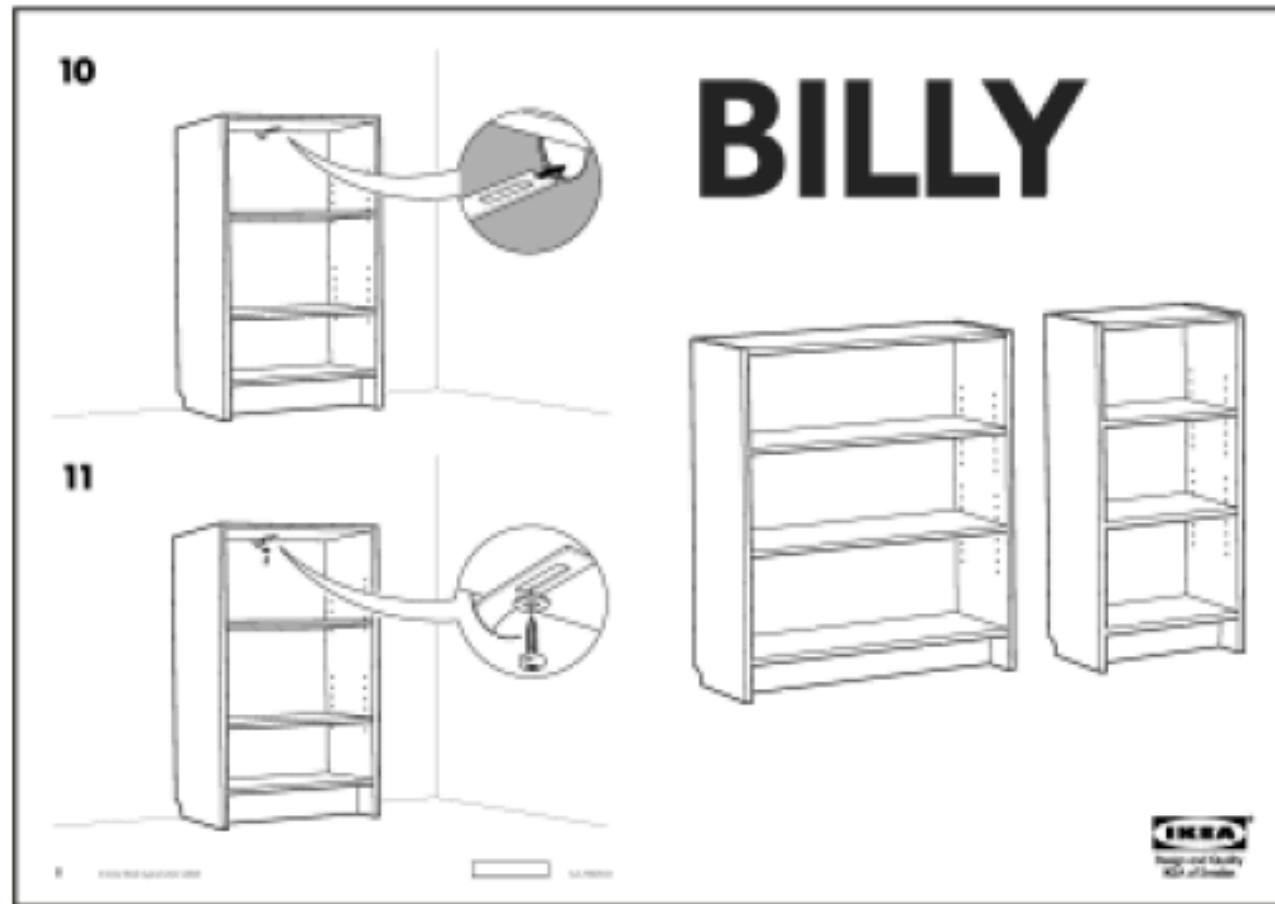
Beispiel: Kochrezept

Algorithmus zur Zubereitung eines Gerichts



Beispiel: Bauanleitung

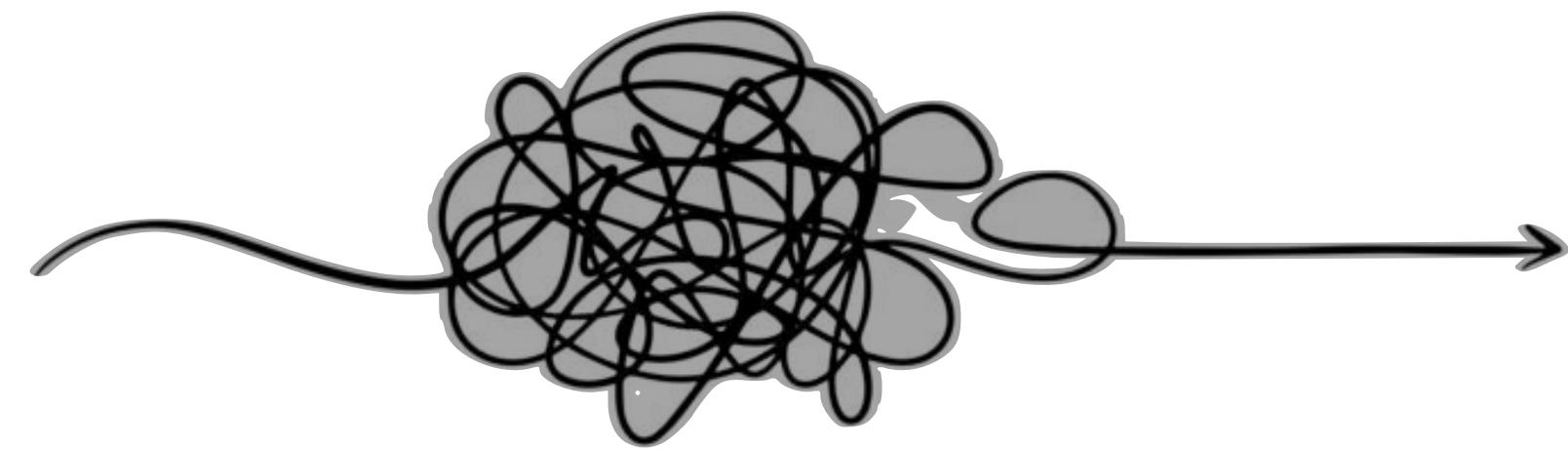
Algorithmus zum Bau eines bestimmten Objekts



Beispiel: Verschlüsselung

Algorithmus zur Verschlüsselung von Daten

Kannst du lesen,
was hier steht?

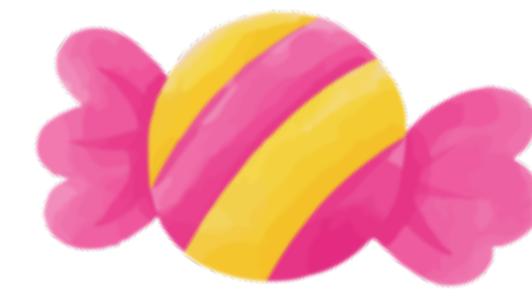
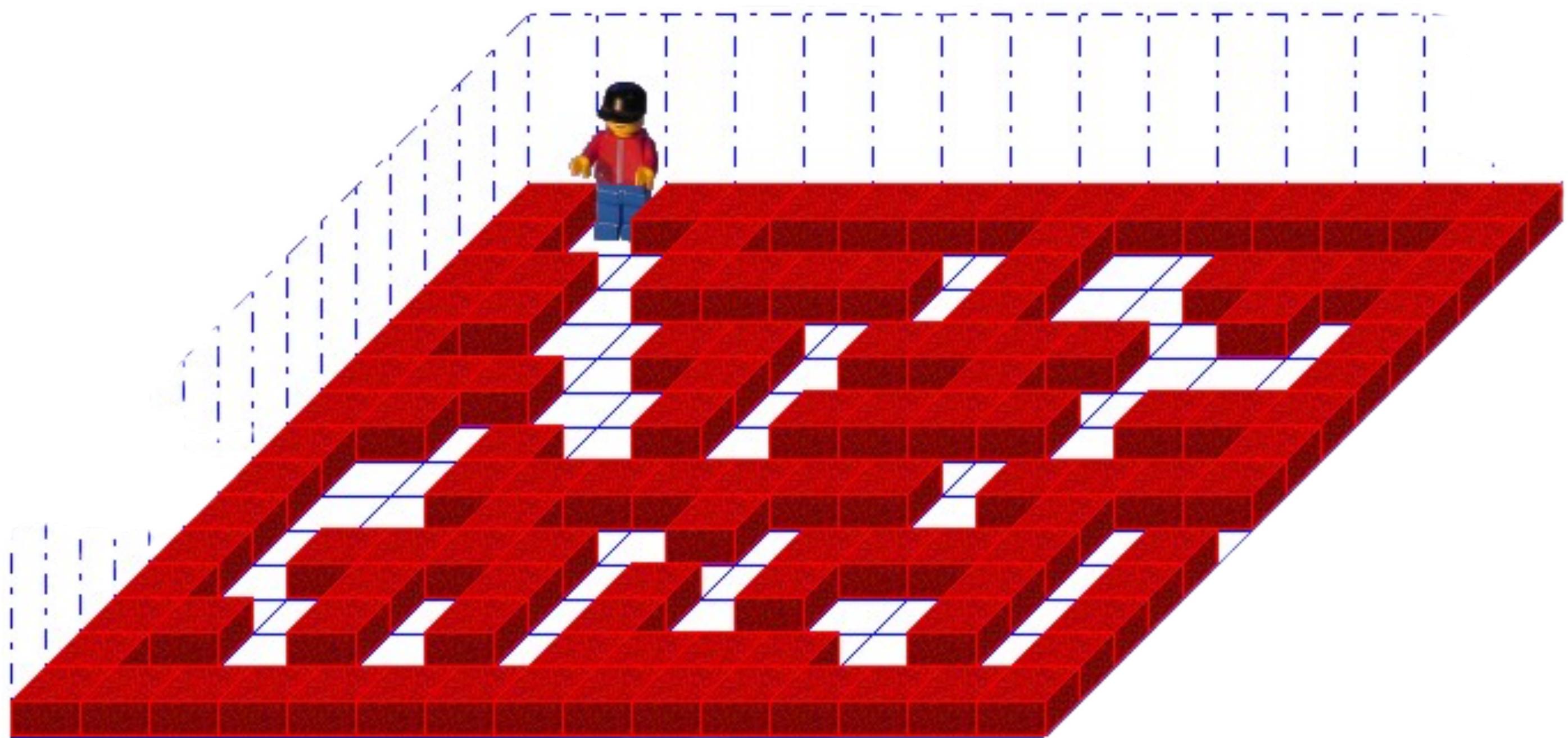


Pfssxy iz qjxjs,
bfx mnjw xyjmy?

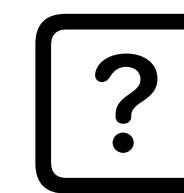
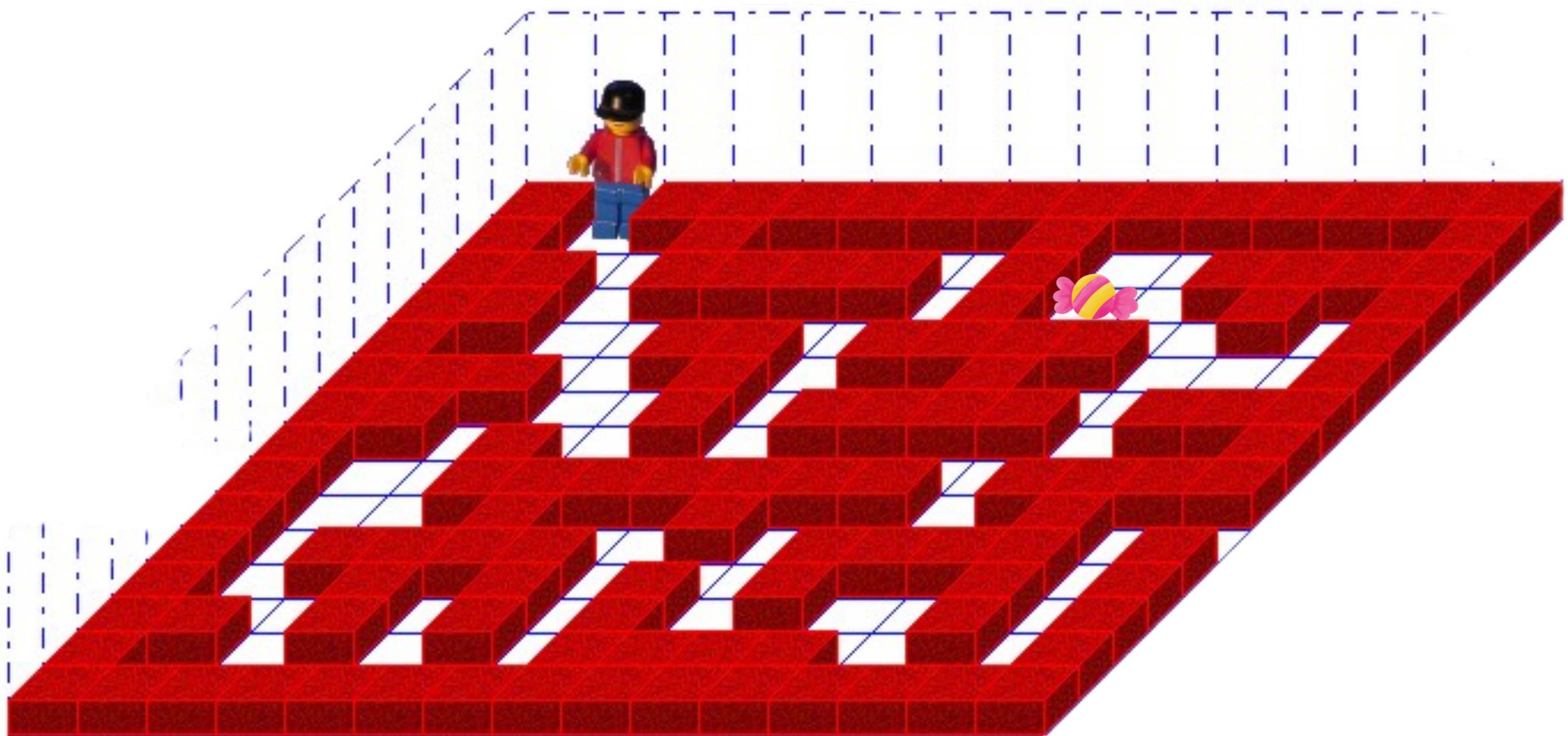
Roboterspiel



Roboterspiel



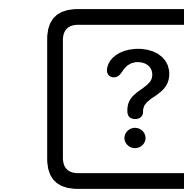
Roboterspiel



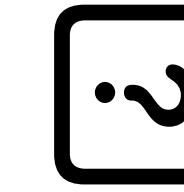
vorwärts



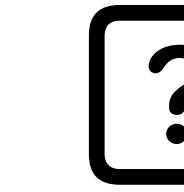
rückwärts



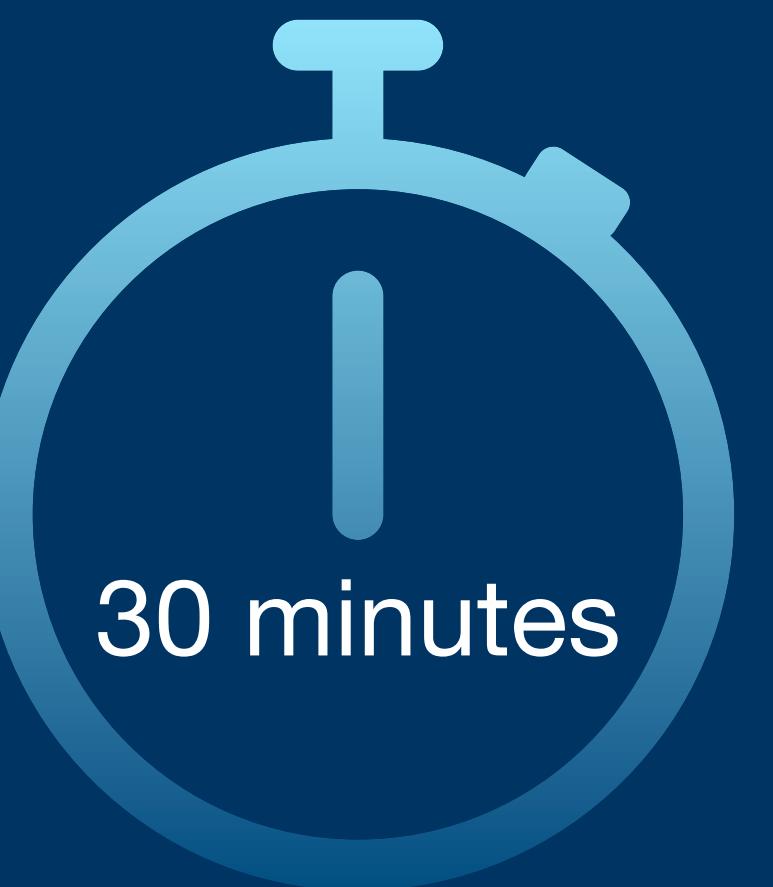
nach links/rechts
drehen



aufheben



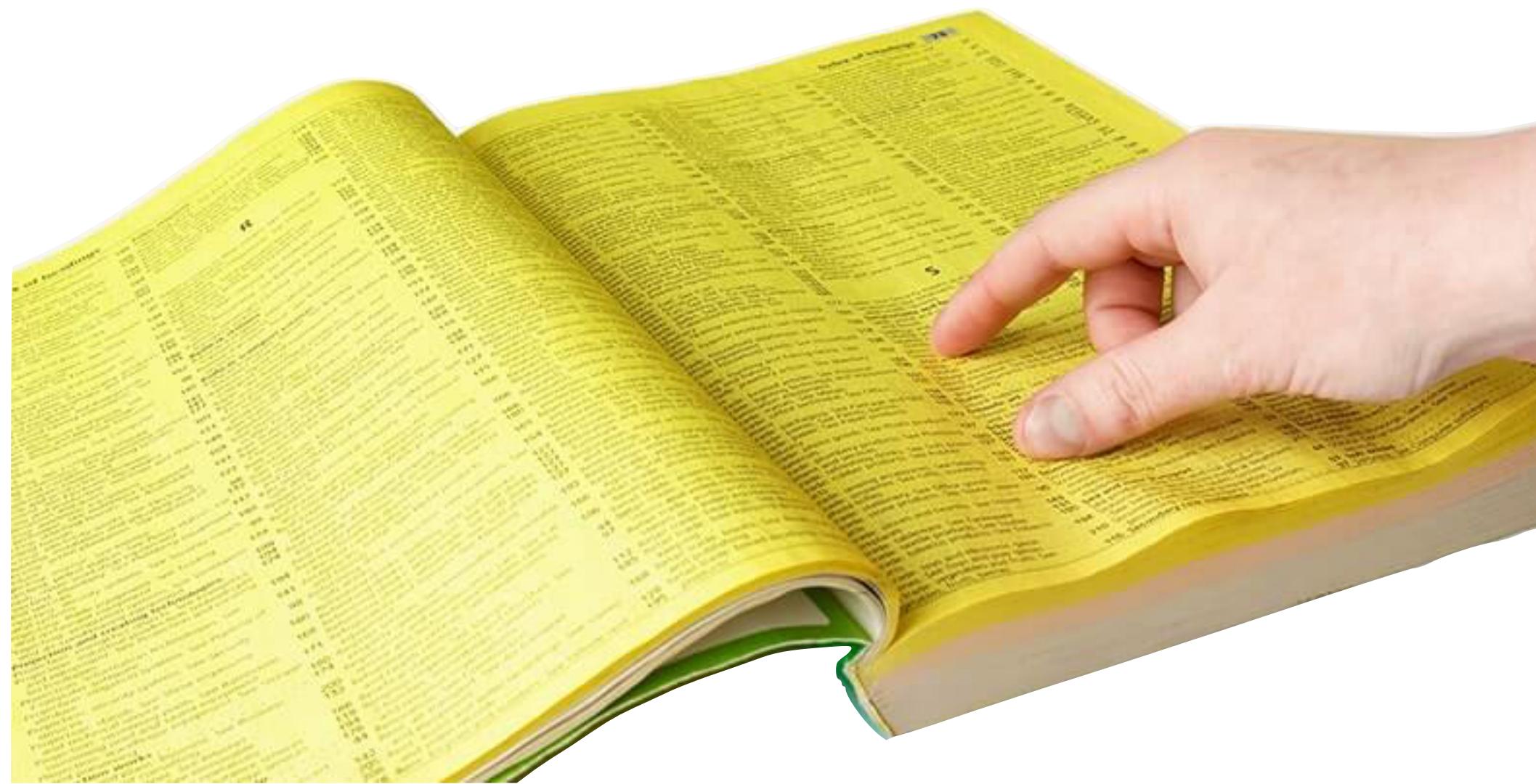
Anweisungen
aufschreiben



Spiel: Roboter

Sortieralgorithmen

Warum brauchen wir Sortieralgorithmen?



ST GMS München

Klasse 3A

Schuljahr: 2018/19
Stand: 08.01.2019

Warum brauchen wir Sortieralgorithmen?

- **Effizienz**
 - Viele Anwendungen benötigen Daten in einer bestimmten Reihenfolge
 - z.B. Telefonbücher, Bibliothekskataloge, ...
- **Optimierung**
 - Sortierte Daten ermöglichen schnellere Such- und Verarbeitungsprozesse

Bubble Sort im Detail

Funktionsweise:

1. Vergleiche benachbarte Elemente und vertausche sie, wenn sie in der falschen Reihenfolge sind
2. Wiederhole den Prozess für jedes Element in der Liste
3. Die größten Elemente "blubbern" nach oben, daher der Name "Bubble Sort"

Sortieralgorithmen

Bubble Sort

5

1

4

9

0

8

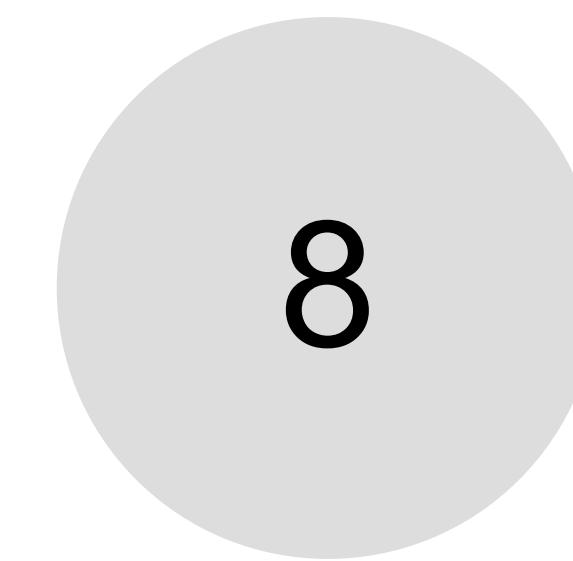
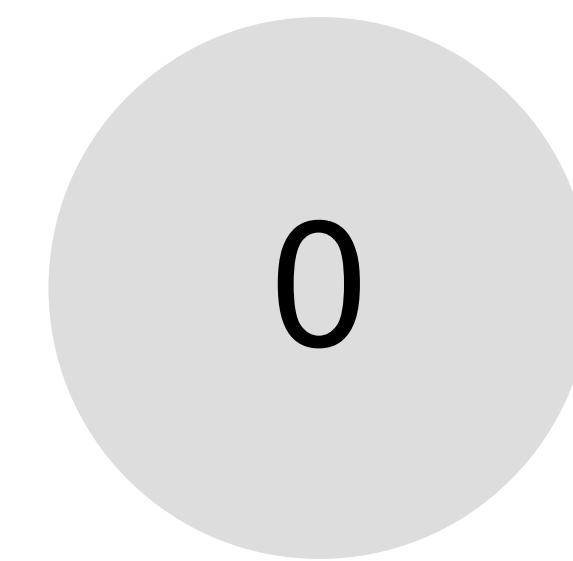
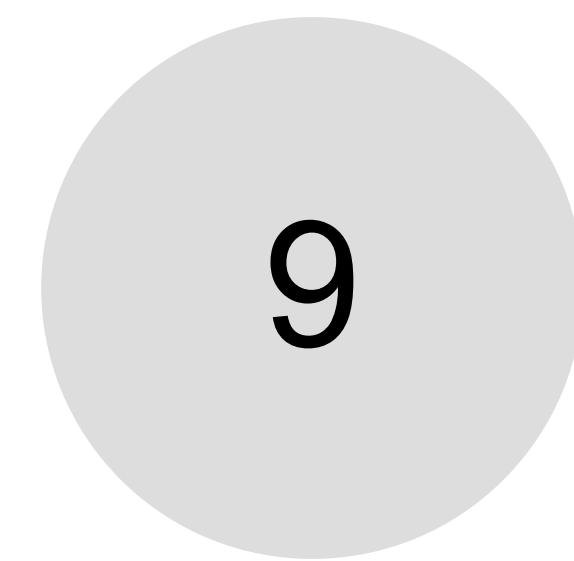
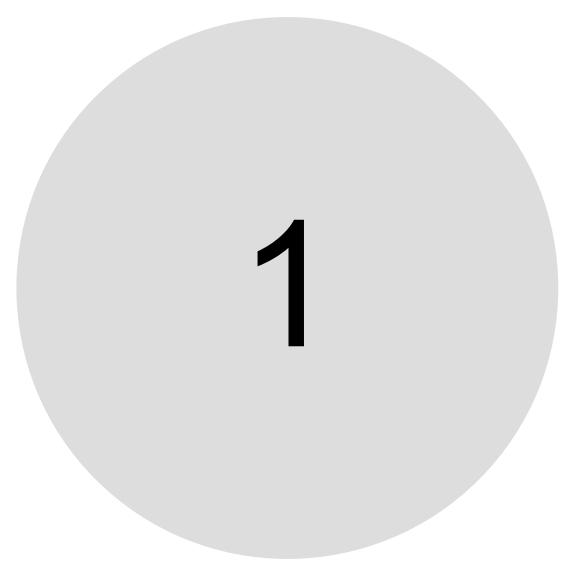
Sortieralgorithmen

Bubble Sort



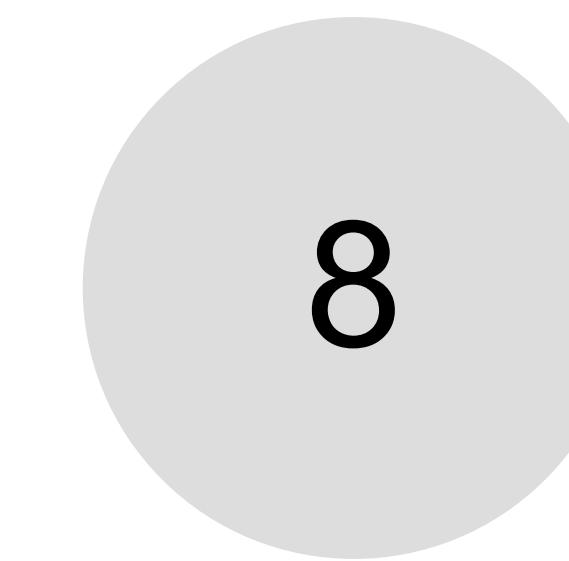
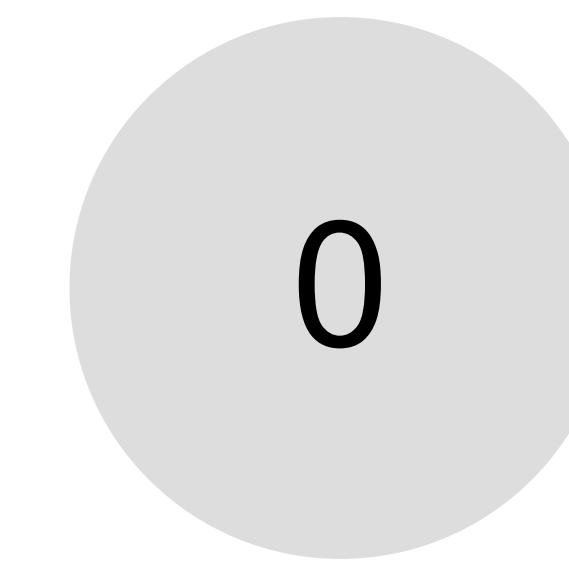
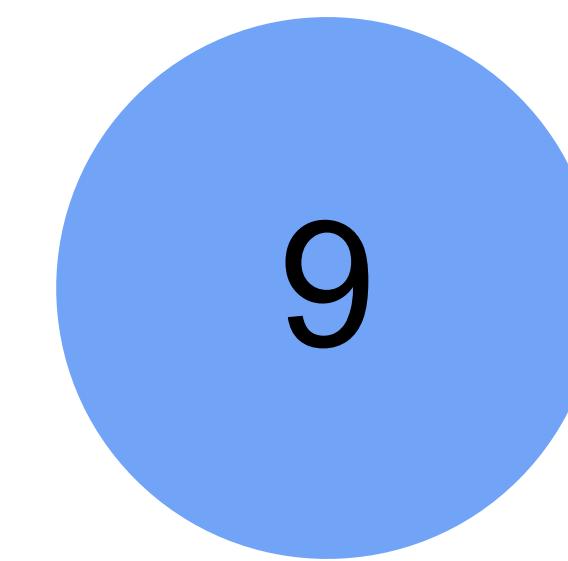
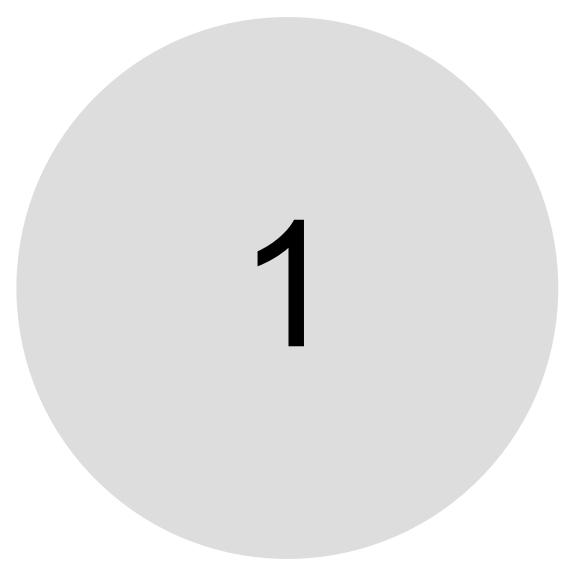
Sortieralgorithmen

Bubble Sort



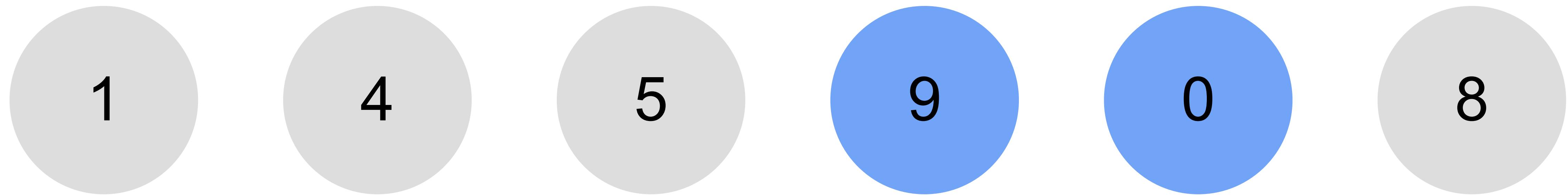
Sortieralgorithmen

Bubble Sort



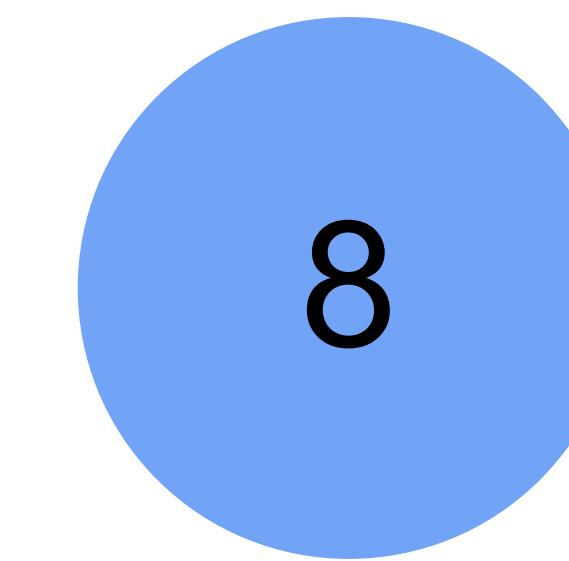
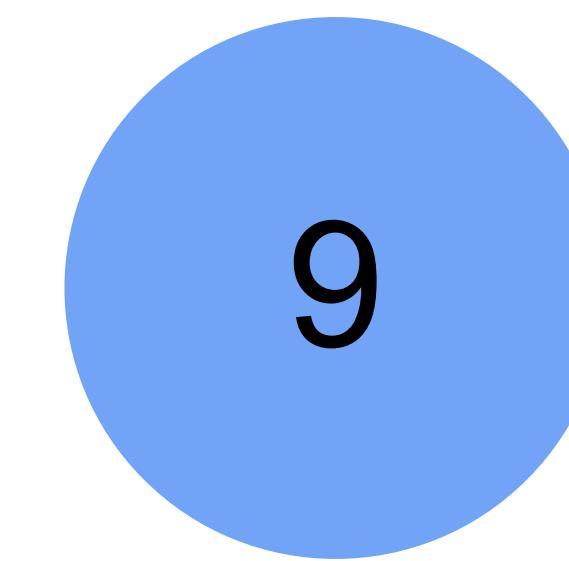
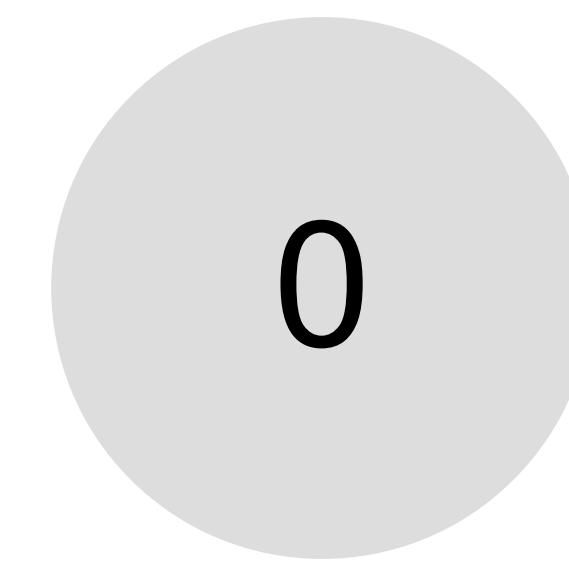
Sortieralgorithmen

Bubble Sort



Sortieralgorithmen

Bubble Sort



Sortieralgorithmen

Bubble Sort



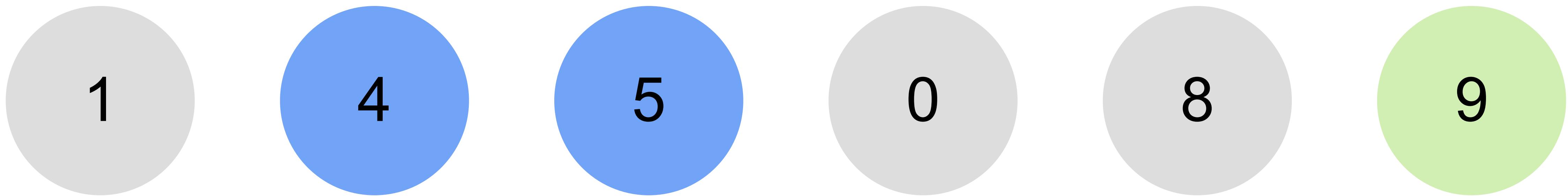
Sortieralgorithmen

Bubble Sort



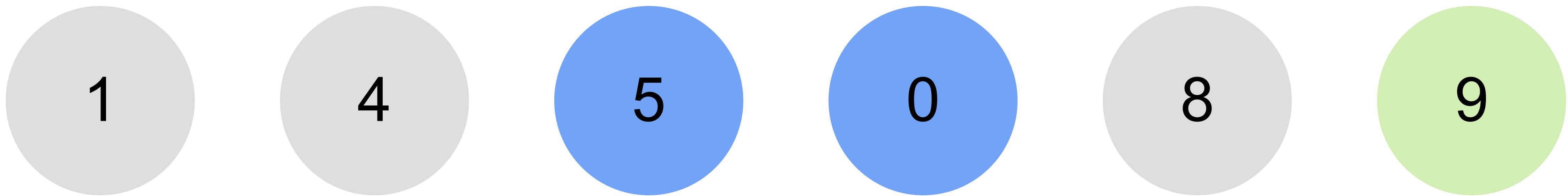
Sortieralgorithmen

Bubble Sort



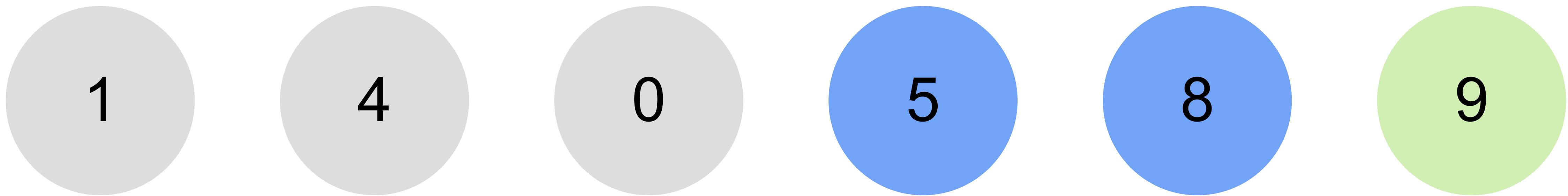
Sortieralgorithmen

Bubble Sort



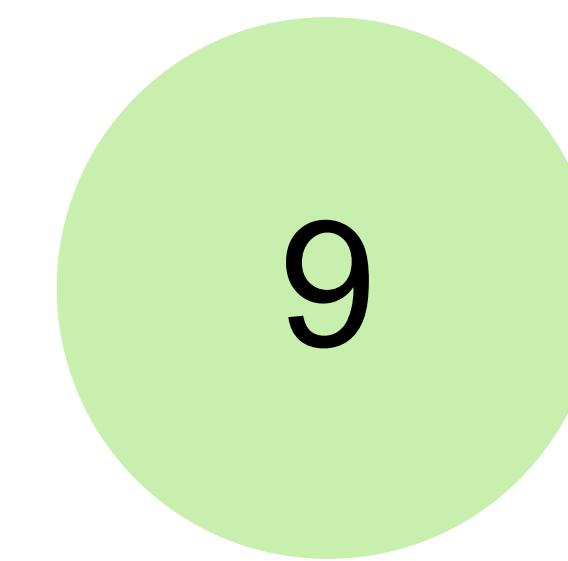
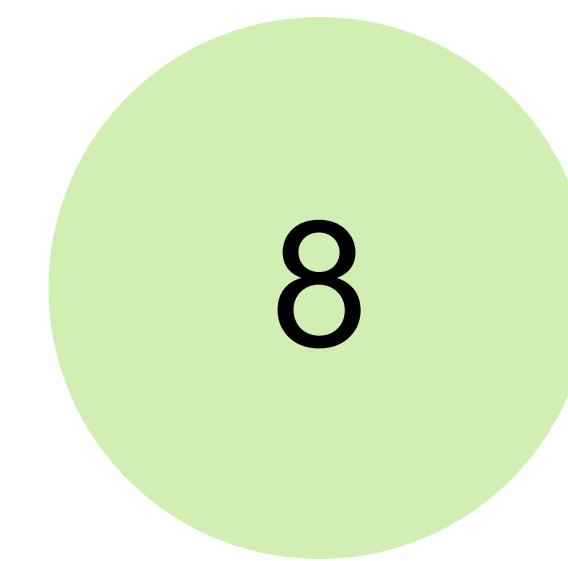
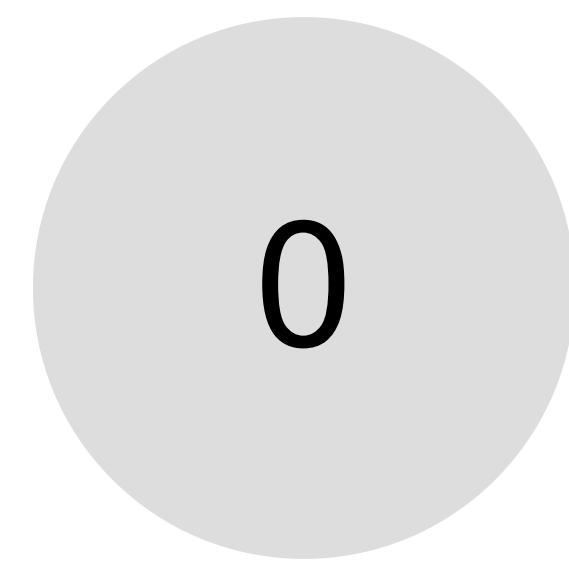
Sortieralgorithmen

Bubble Sort



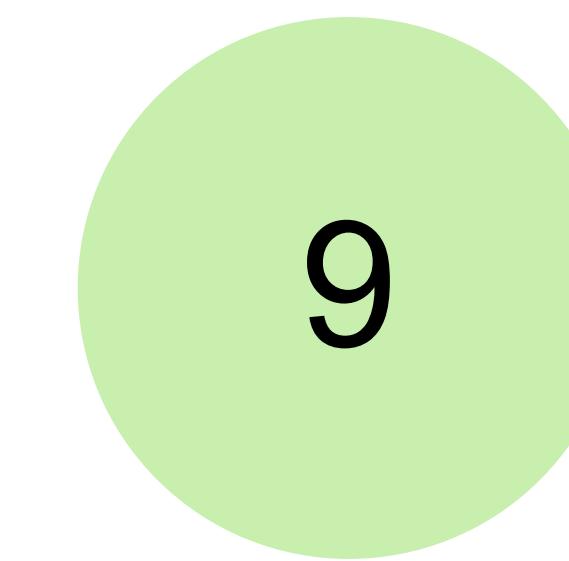
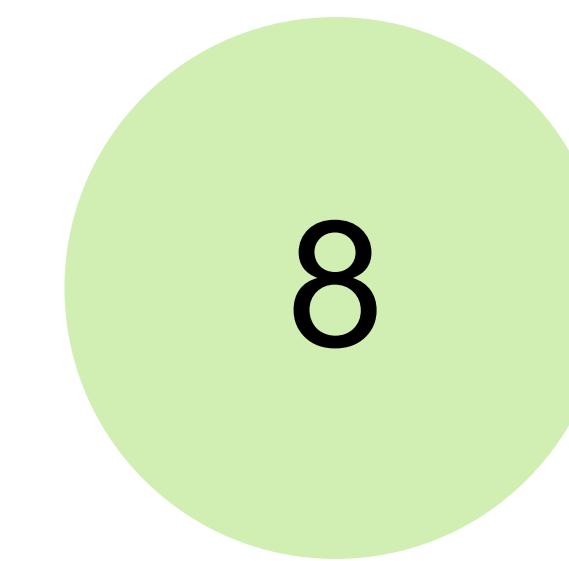
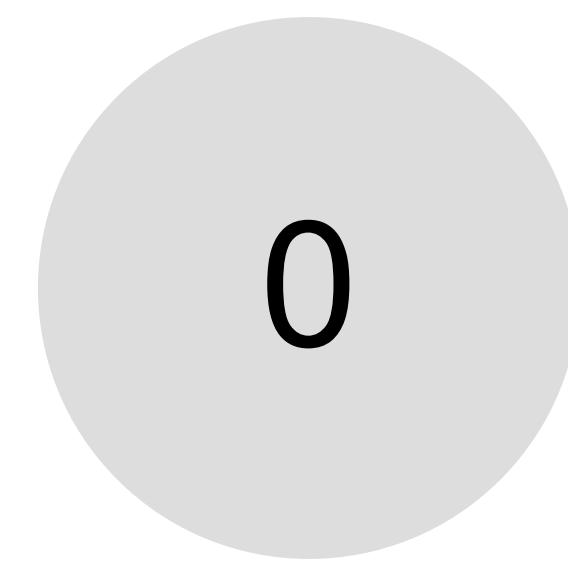
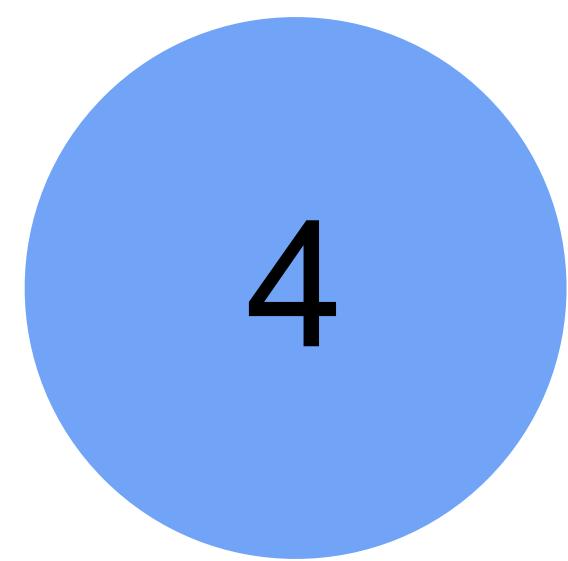
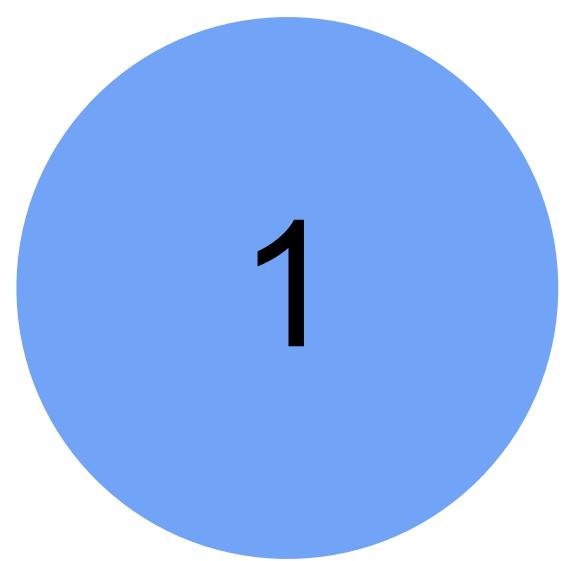
Sortieralgorithmen

Bubble Sort



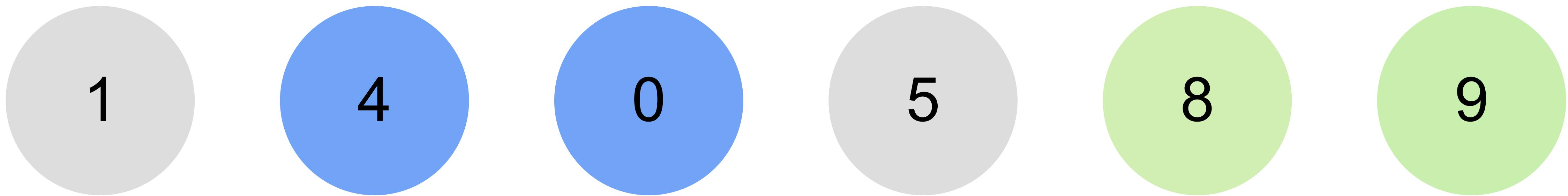
Sortieralgorithmen

Bubble Sort



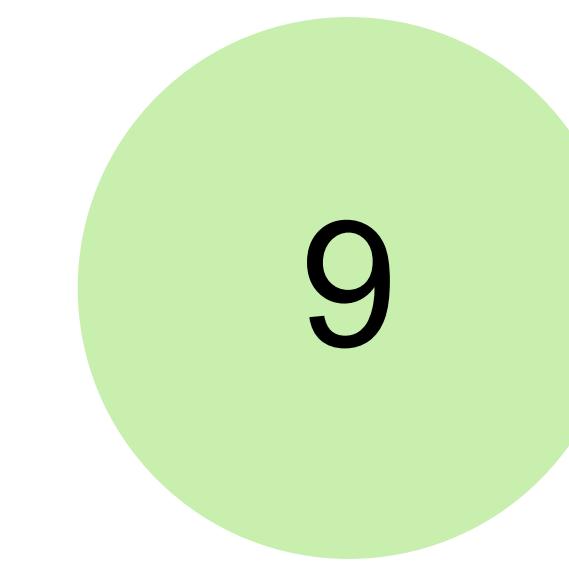
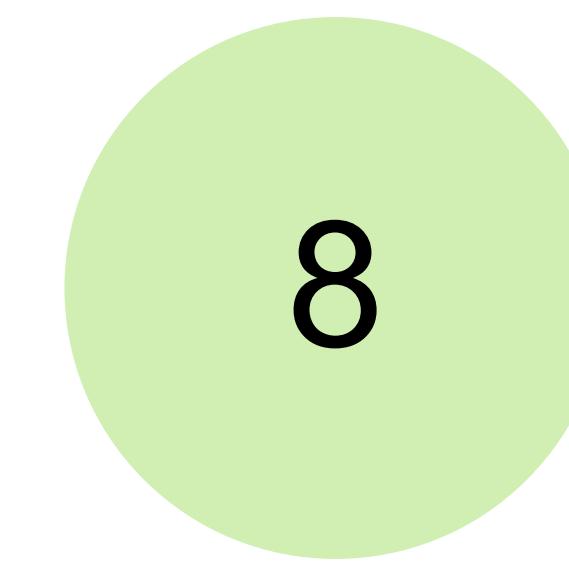
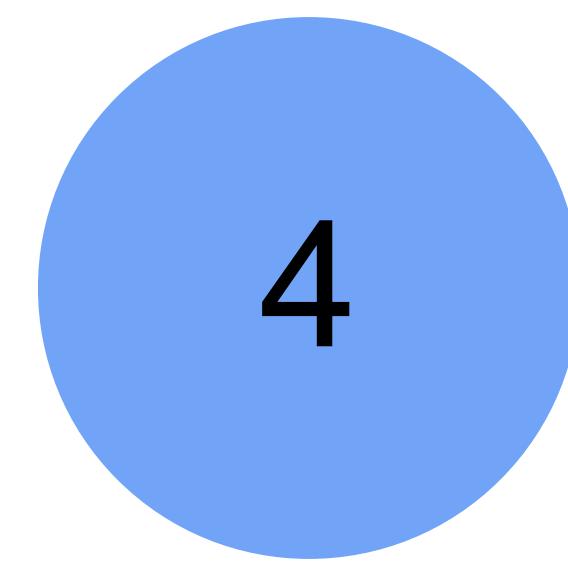
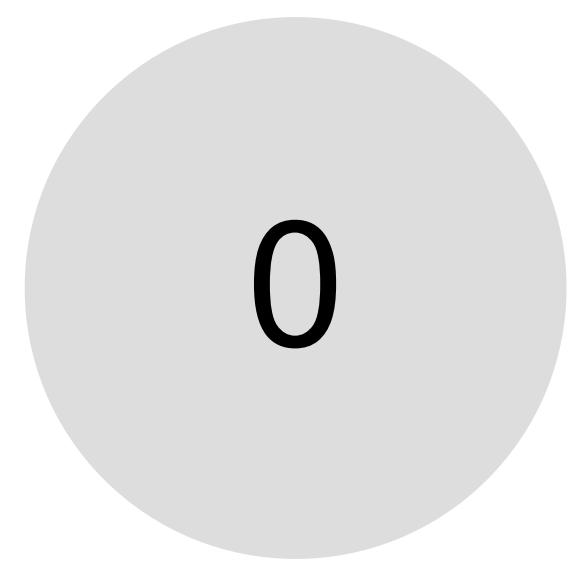
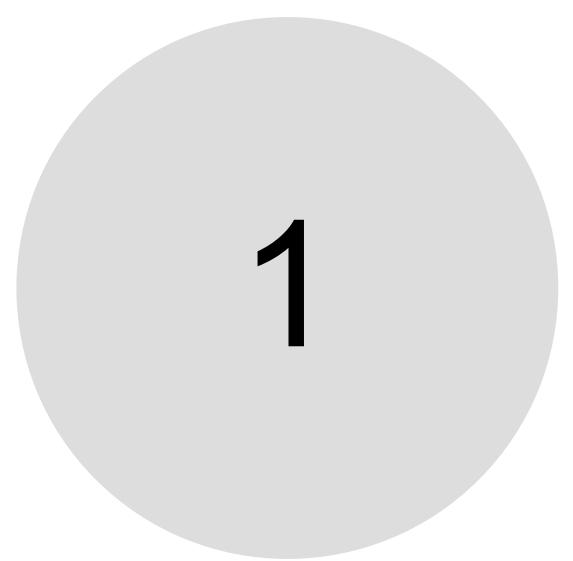
Sortieralgorithmen

Bubble Sort



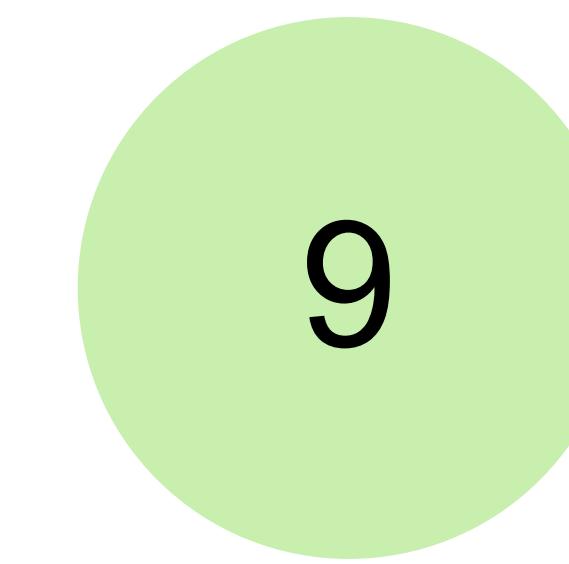
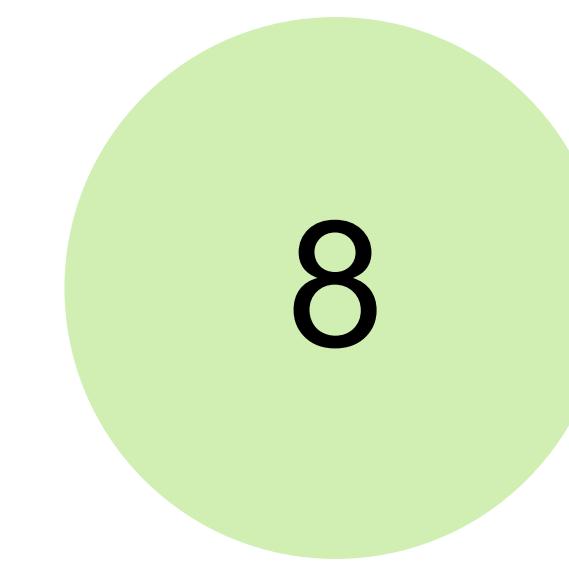
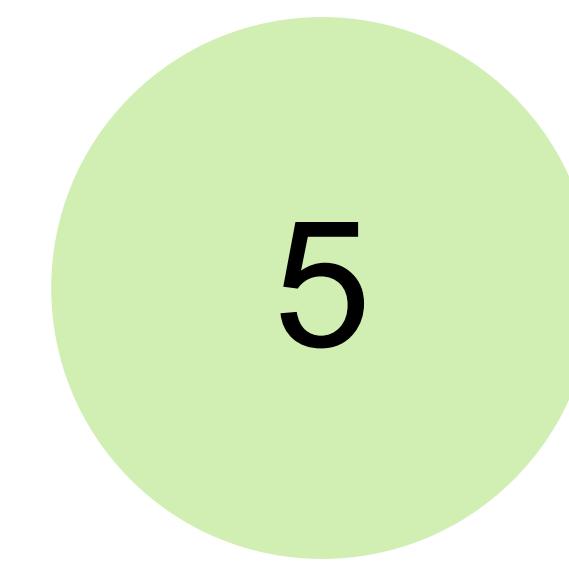
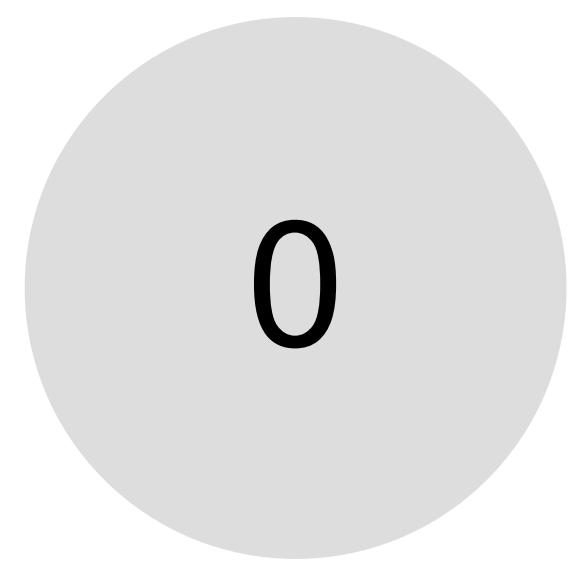
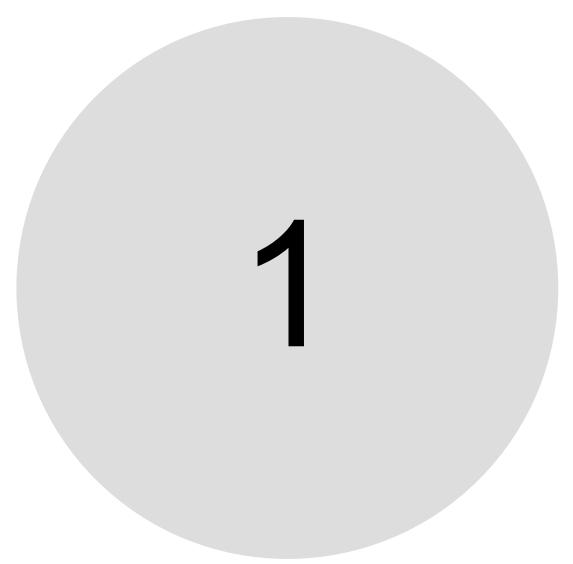
Sortieralgorithmen

Bubble Sort



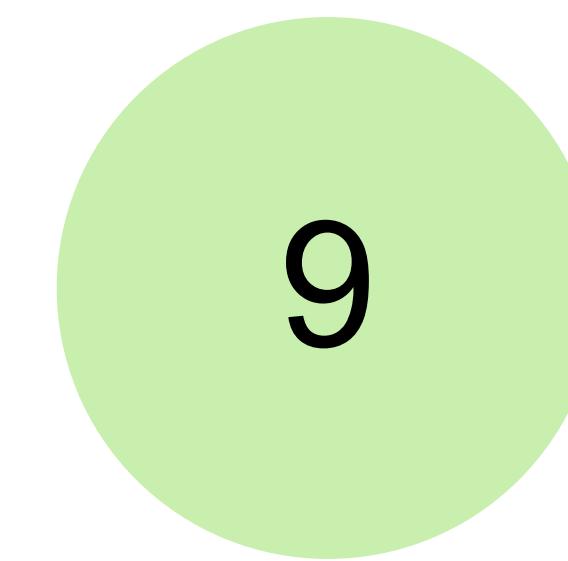
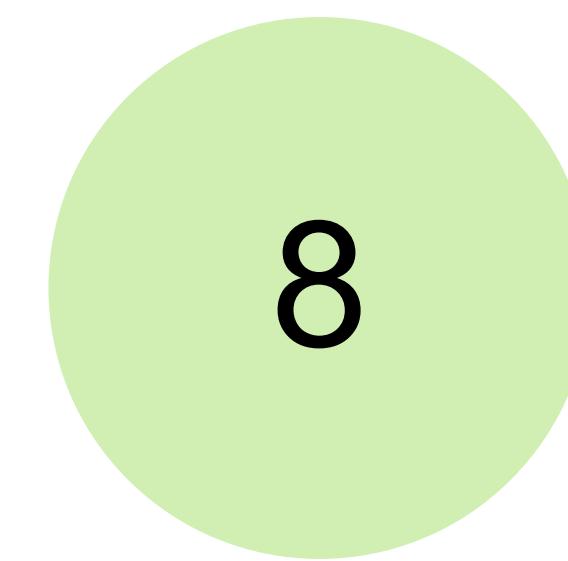
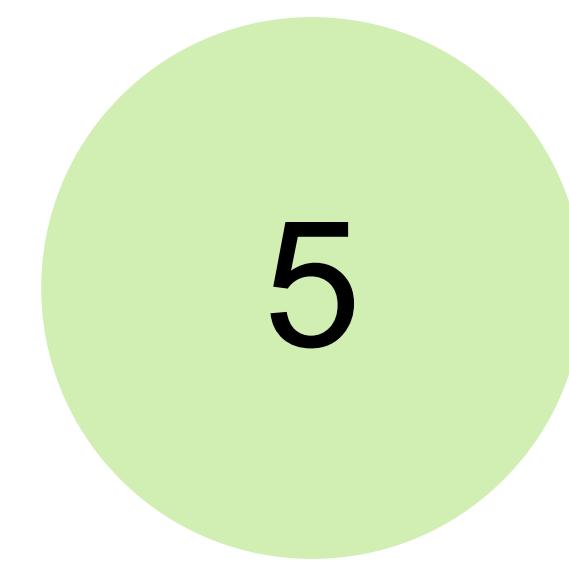
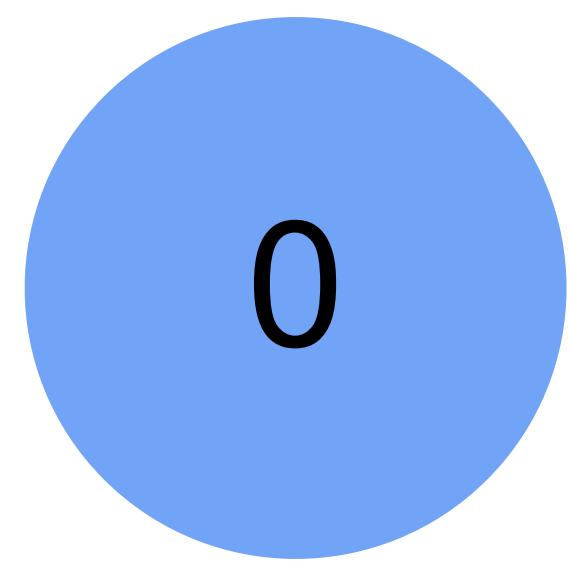
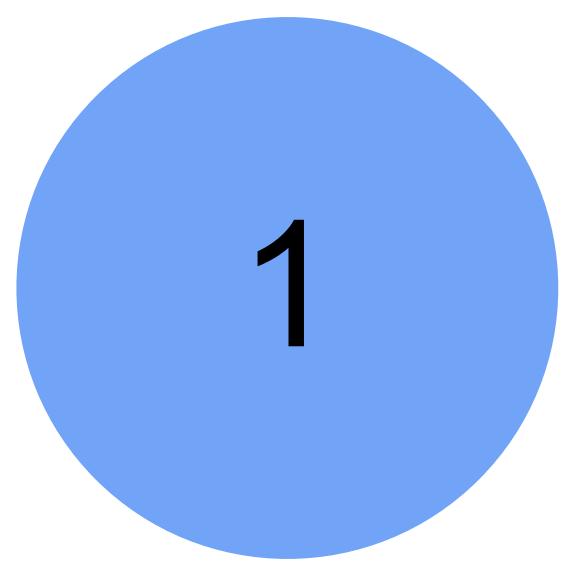
Sortieralgorithmen

Bubble Sort



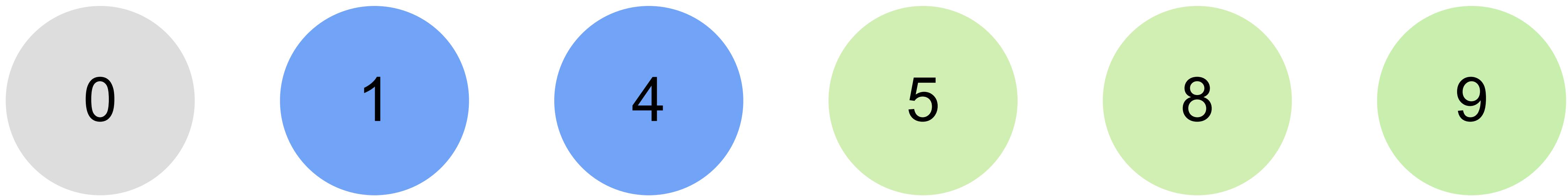
Sortieralgorithmen

Bubble Sort



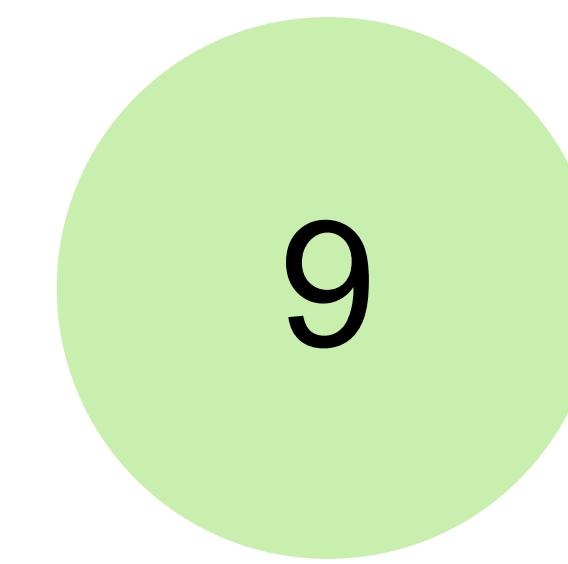
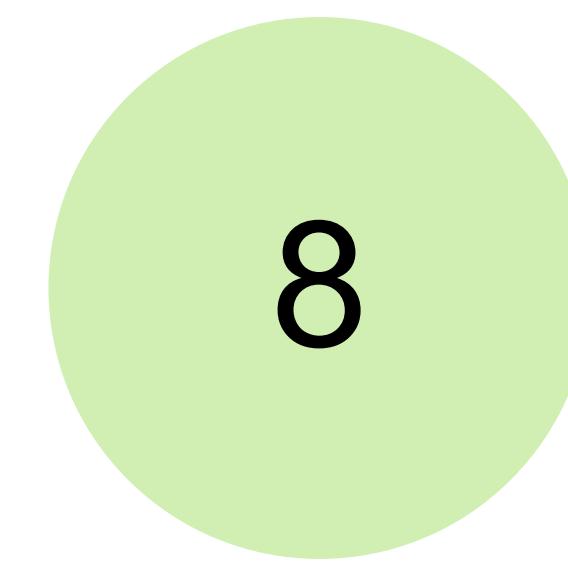
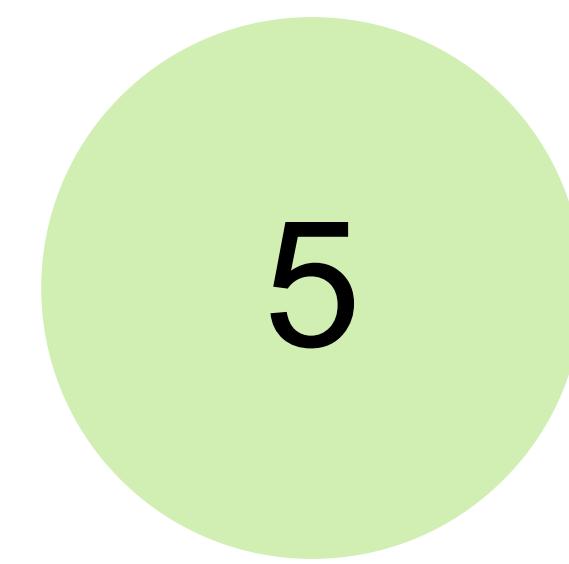
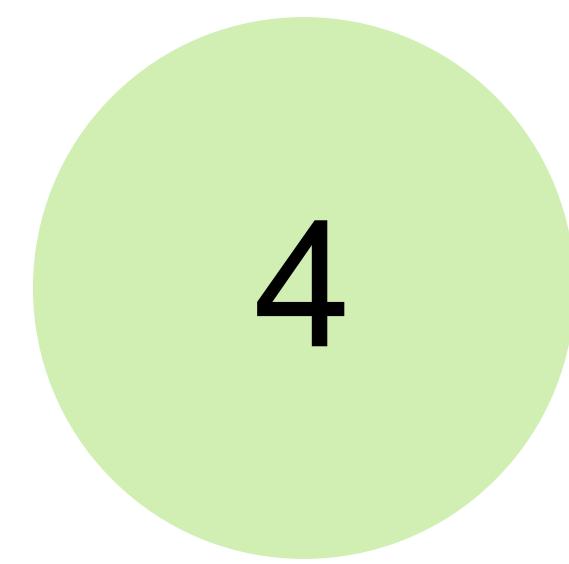
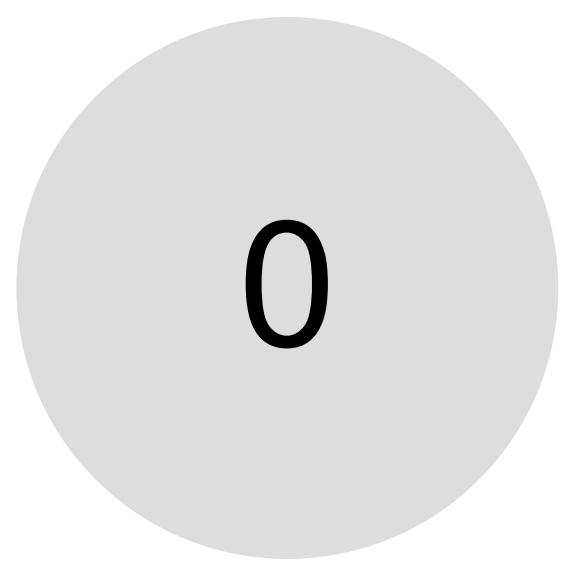
Sortieralgorithmen

Bubble Sort



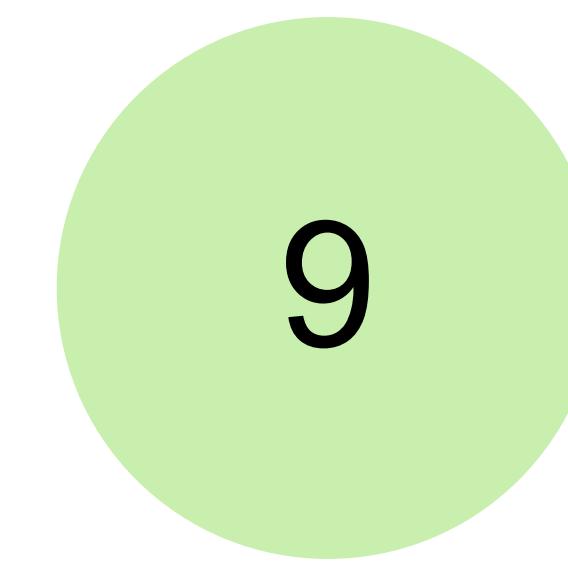
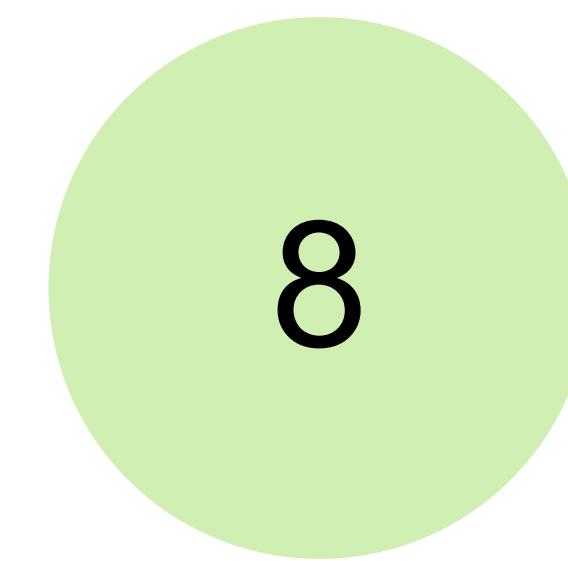
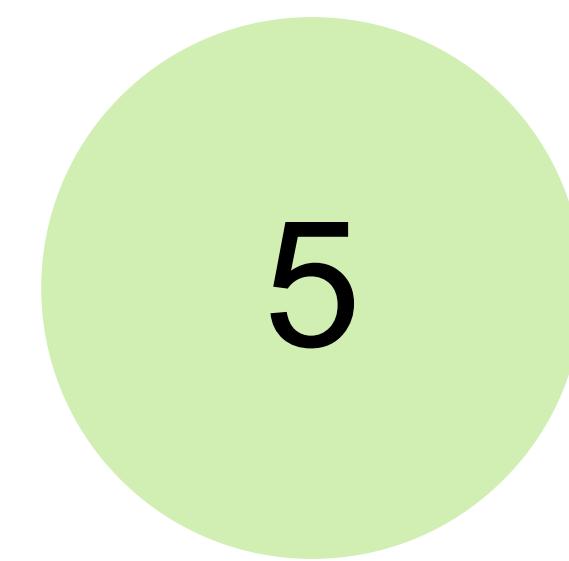
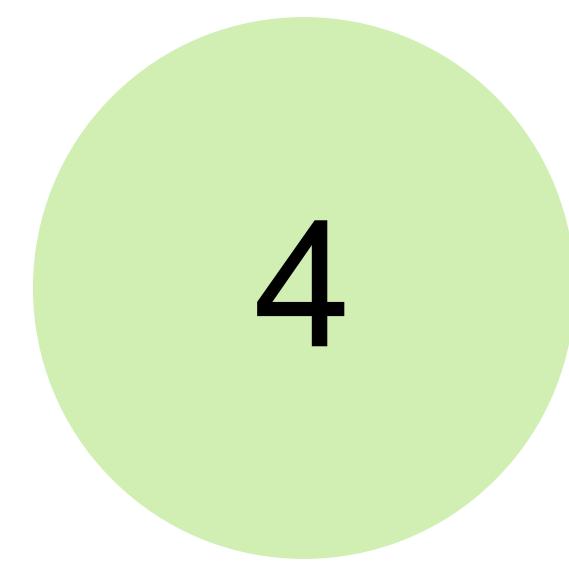
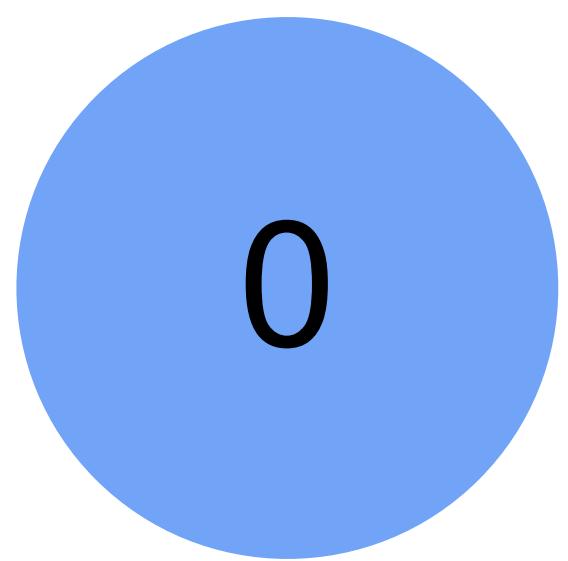
Sortieralgorithmen

Bubble Sort



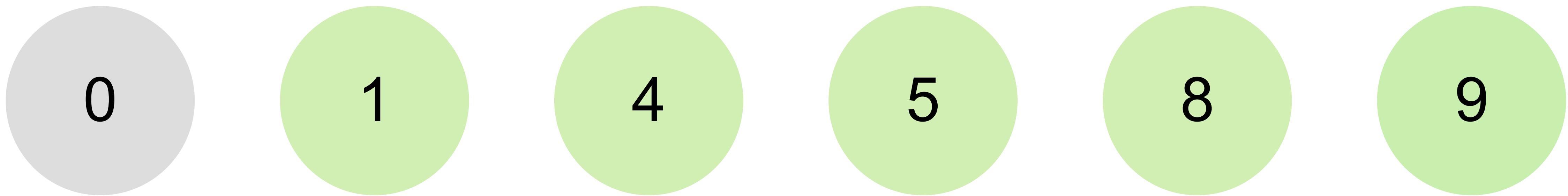
Sortieralgorithmen

Bubble Sort



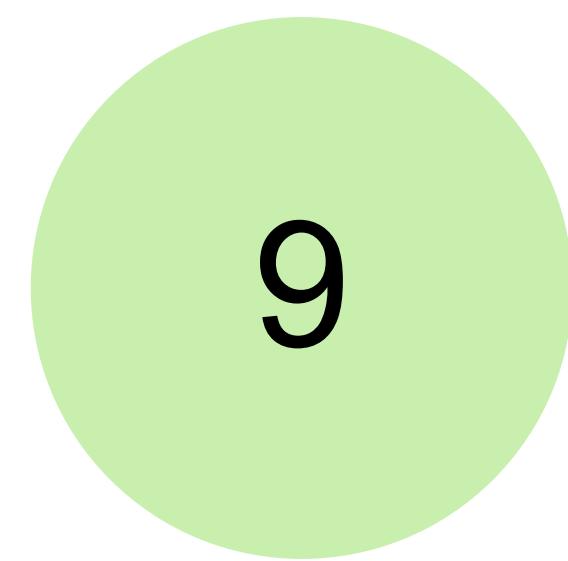
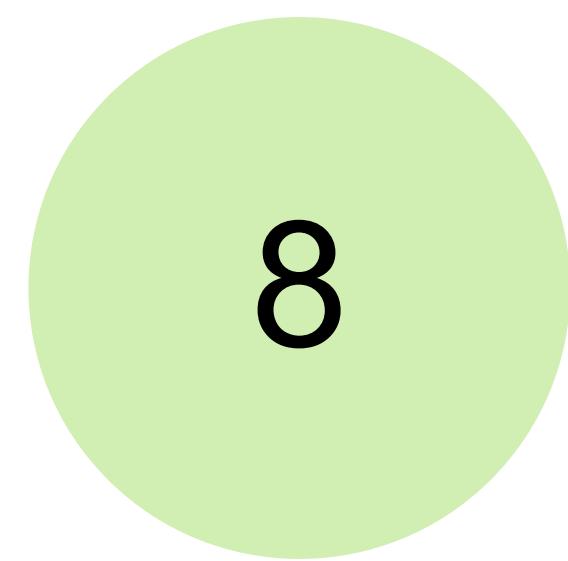
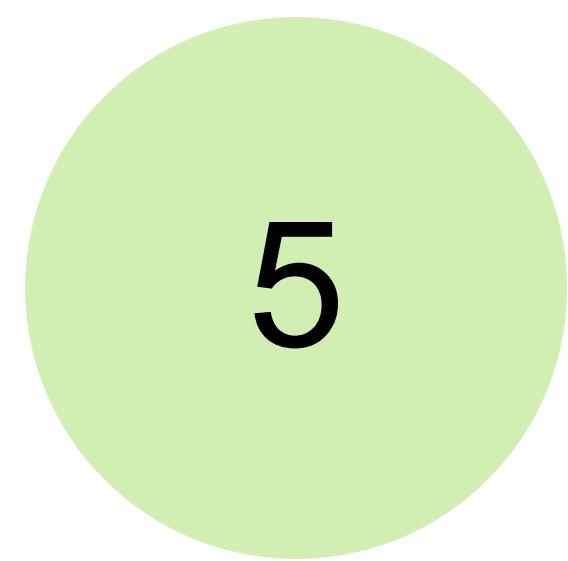
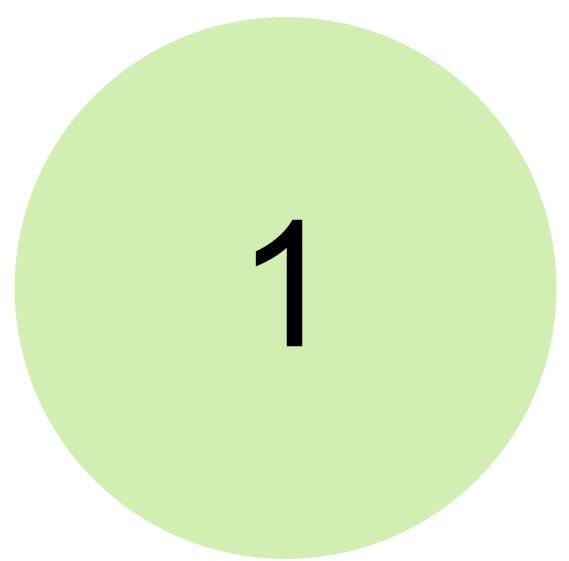
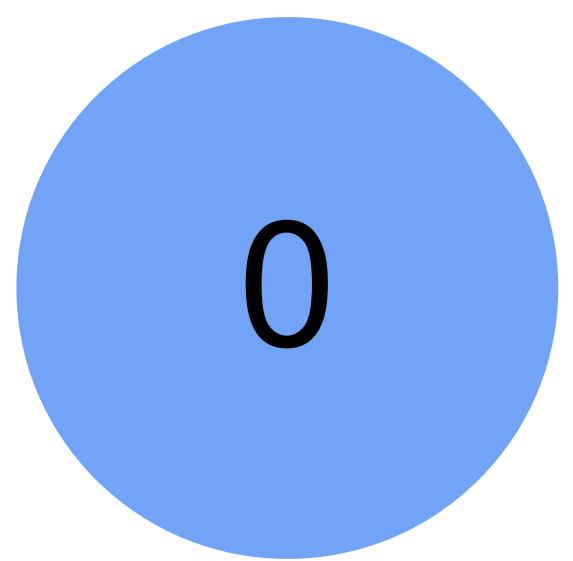
Sortieralgorithmen

Bubble Sort



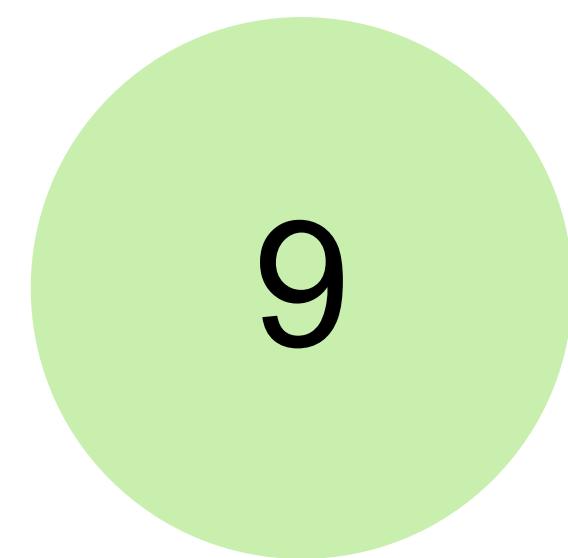
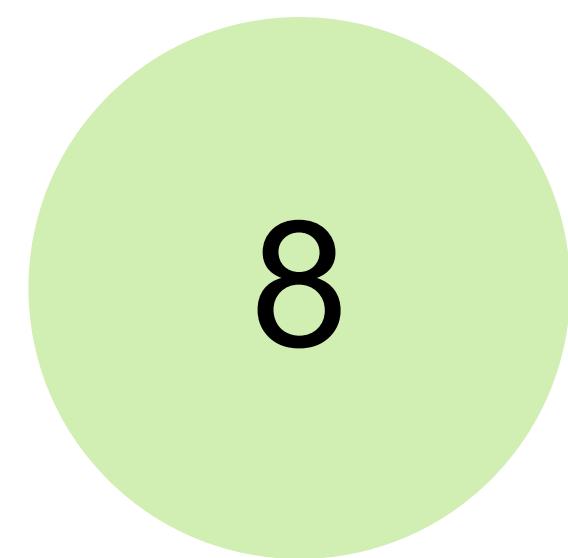
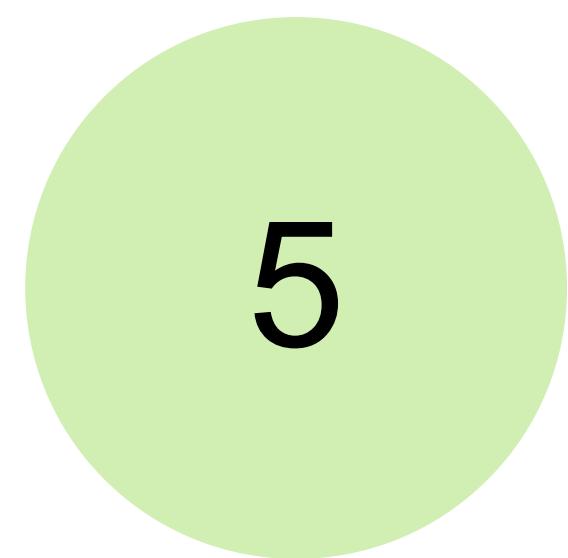
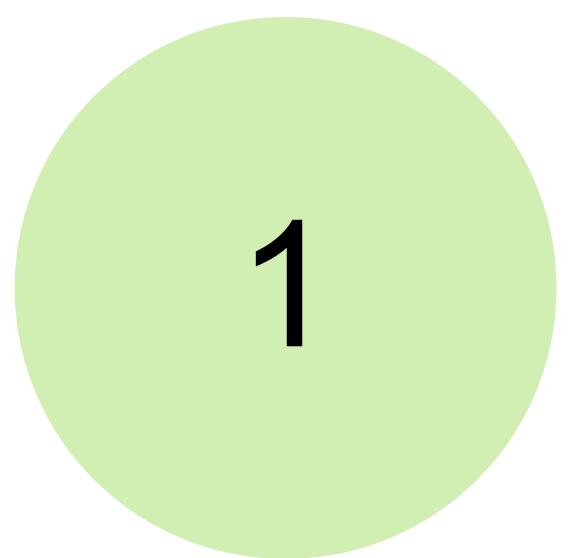
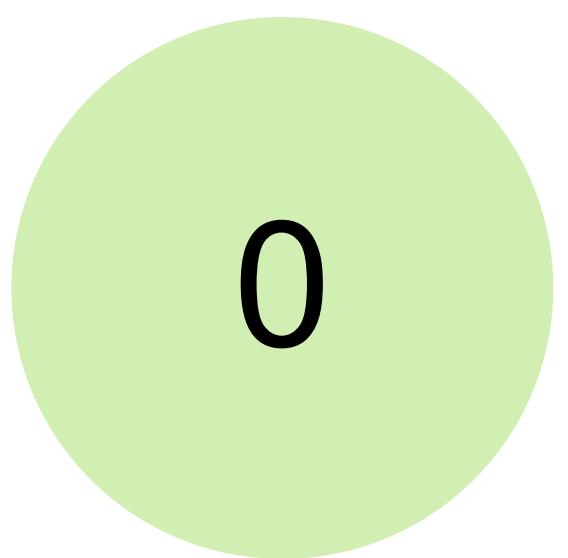
Sortieralgorithmen

Bubble Sort



Sortieralgorithmen

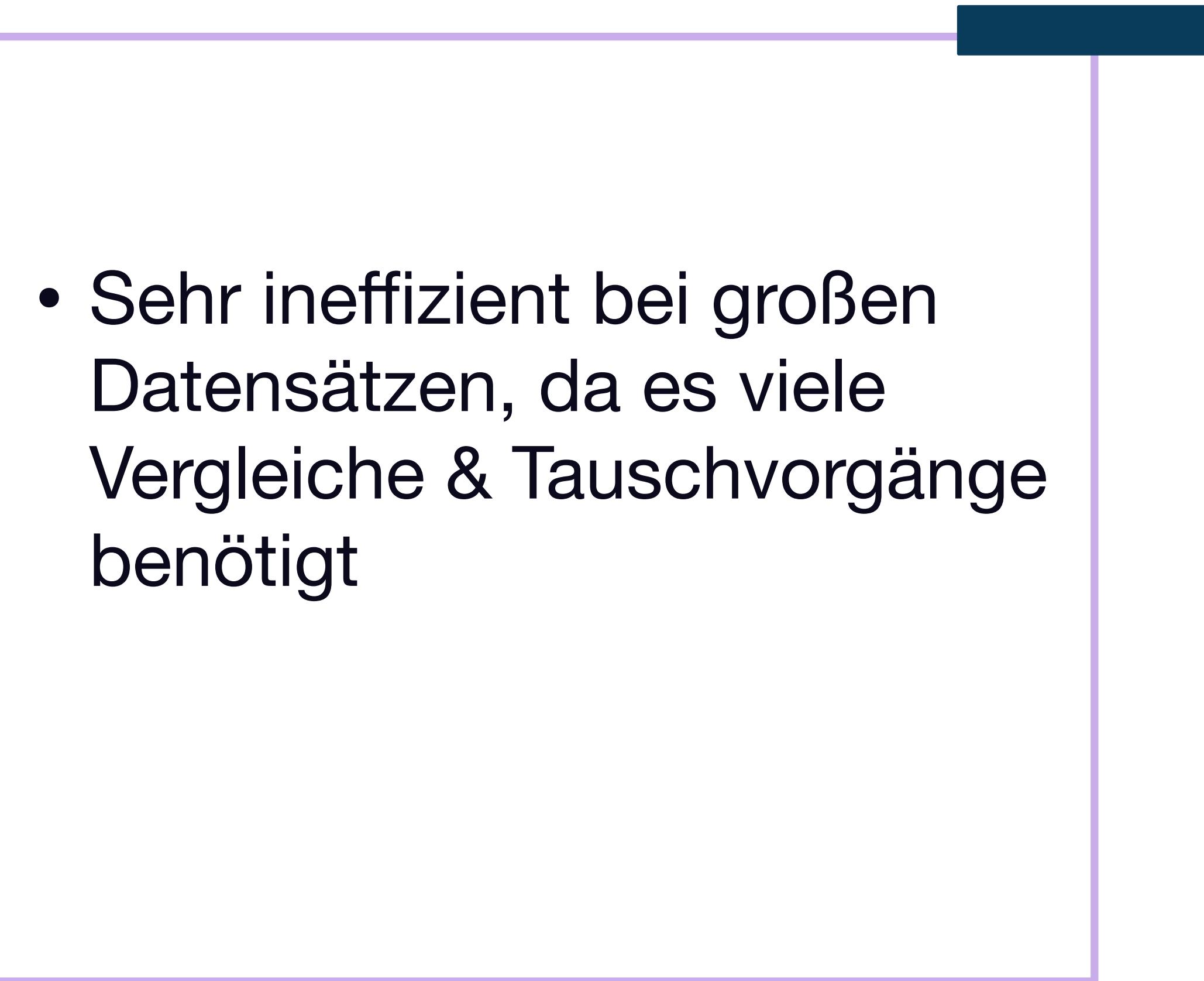
Bubble Sort



Vorteile und Nachteile



- Einfach zu verstehen
- Einfach zu implementieren



- Sehr ineffizient bei großen Datensätzen, da es viele Vergleiche & Tauschvorgänge benötigt

Merge Sort im Detail

Ausgangssituation: Unsortierte Liste von Zahlen

Funktionsweise:

1. Aufteilen (Divide)
 - a. Wir teilen diese Liste immer wieder in der Mitte, bis wir nur noch Listen mit einem einzigen Element haben
 - b. Eine Liste mit einem Element ist immer sortiert
2. Wiederhole den Prozess für jedes Element in der Liste
 - a. Jetzt, wo wir viele kleine Listen mit nur einem Element haben, beginnen wir, diese wieder zusammenzufügen
 - b. Dabei achten wir darauf, dass die Zahlen in der richtigen Reihenfolge sind
3. Zusammenführen (Merge)
 - a. Nimm zwei kleine Listen und vergleiche die ersten Elemente jeder Liste
 - b. Füge das kleinere Element in eine neue Liste ein
 - c. Wiederhole diesen Vorgang, bis alle Elemente sortiert in der neuen Liste sind
 - d. Mach das für alle Paare von Listen, bis alle kleinen Listen zu einer großen, sortierten Liste zusammengefügt sind

Sortieralgorithmen

Merge Sort

5

1

4

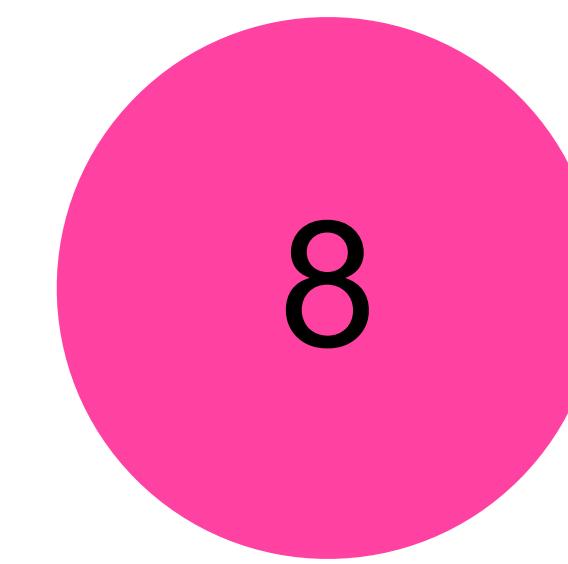
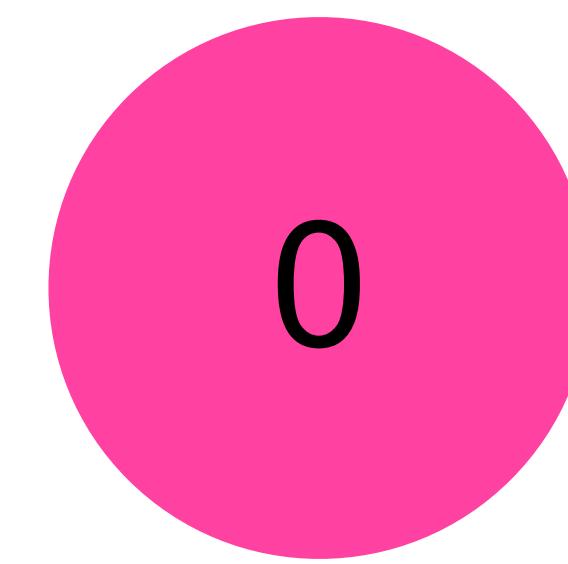
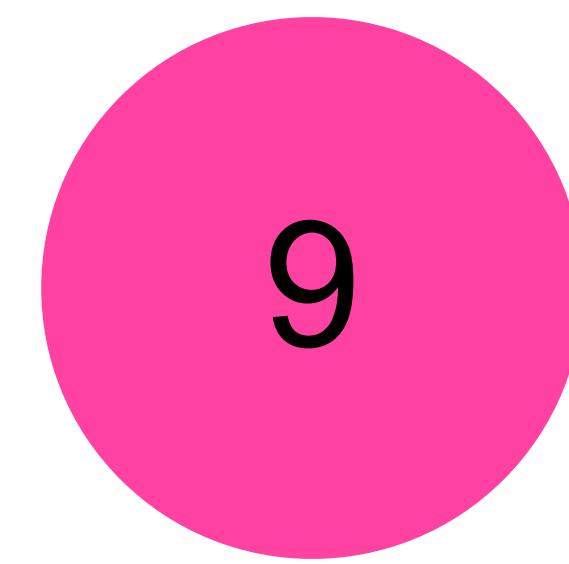
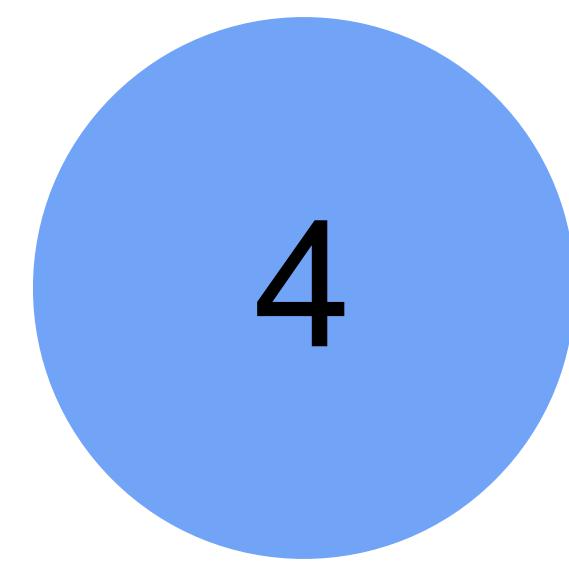
9

0

8

Sortieralgorithmen

Merge Sort



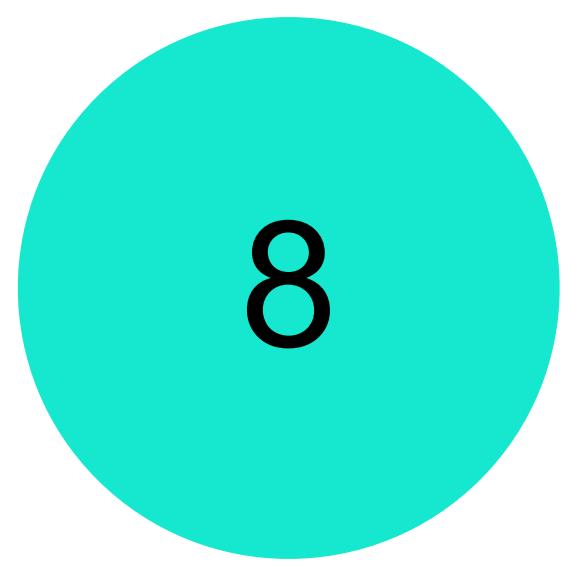
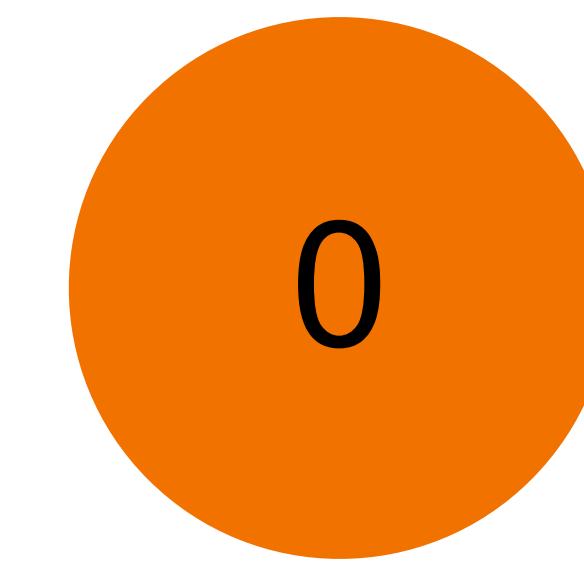
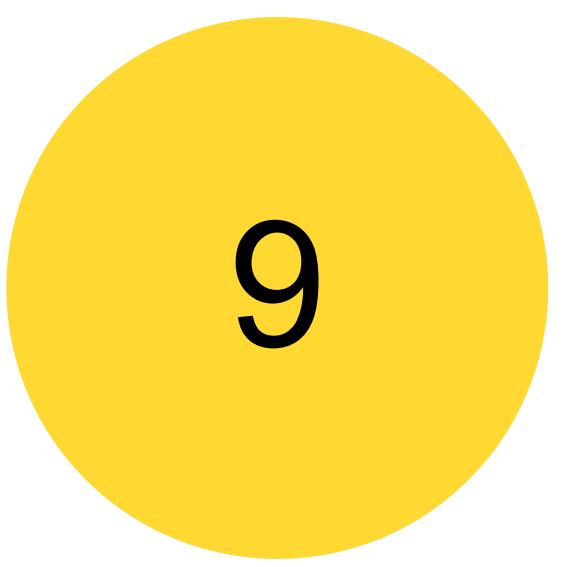
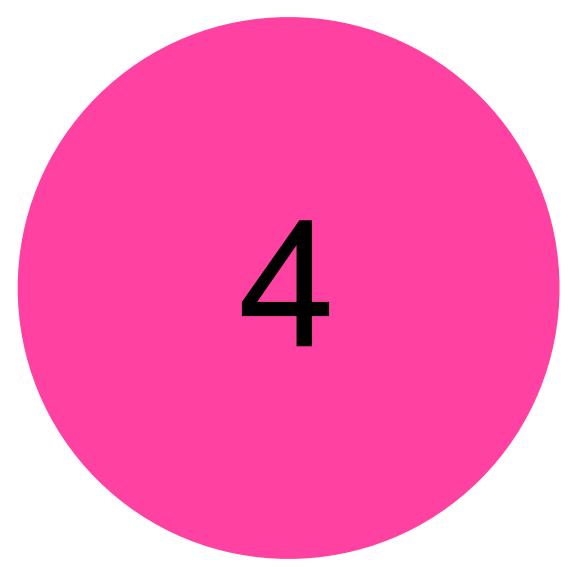
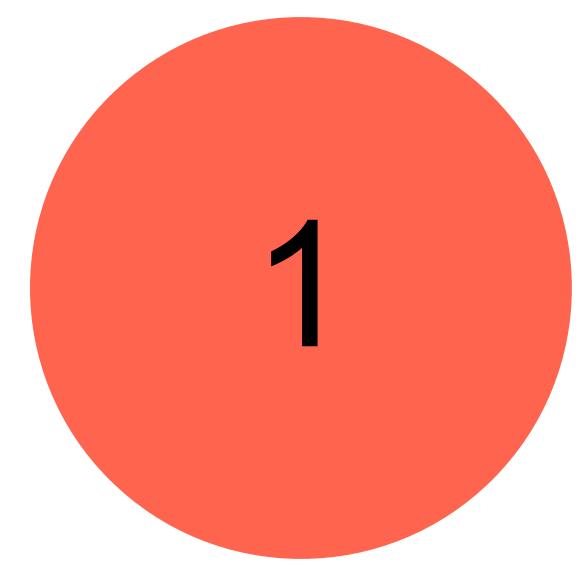
Sortieralgorithmen

Merge Sort



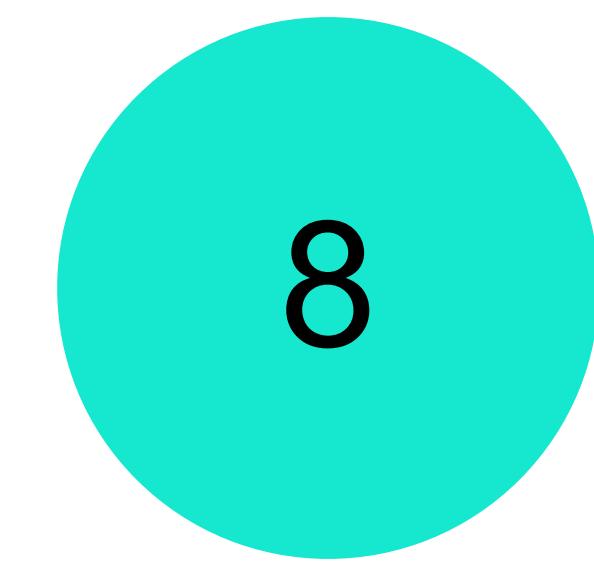
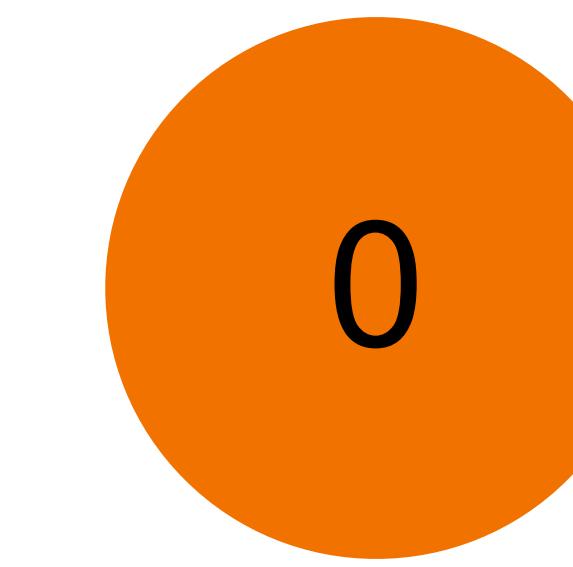
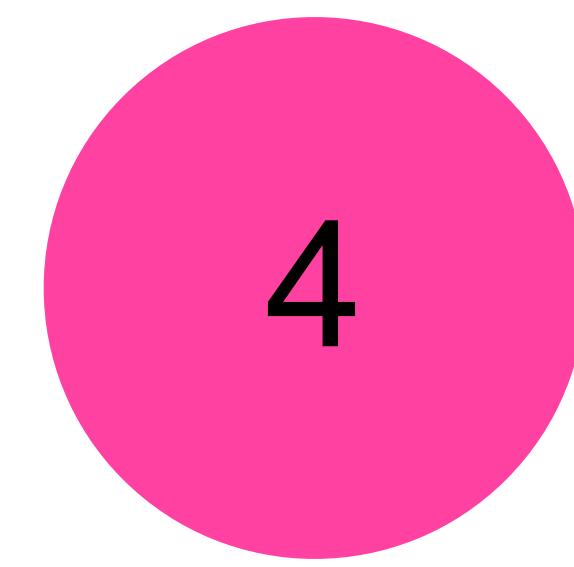
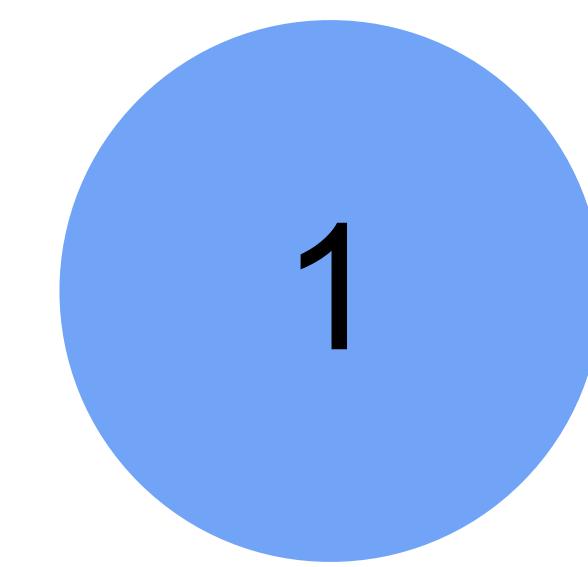
Sortieralgorithmen

Merge Sort



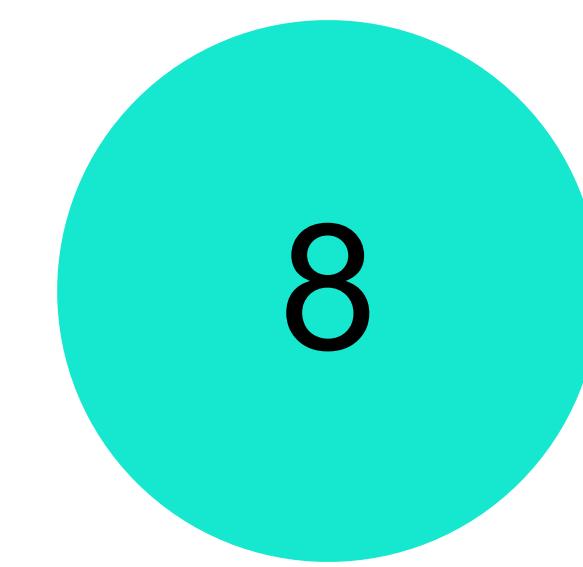
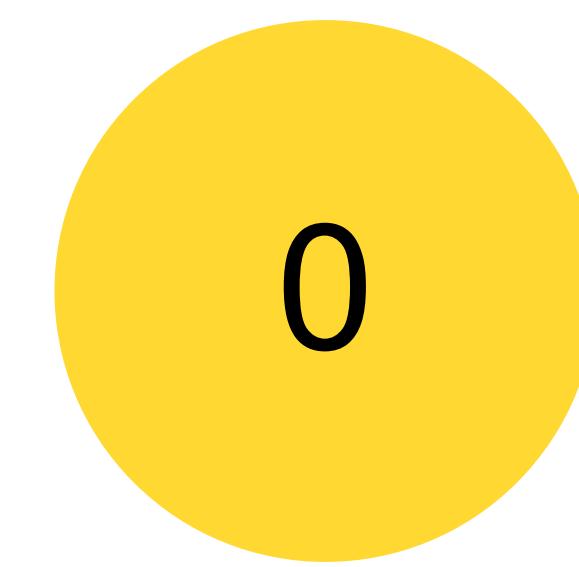
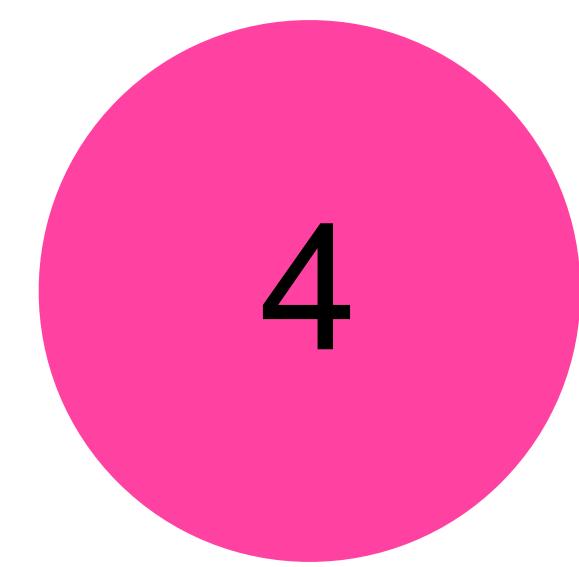
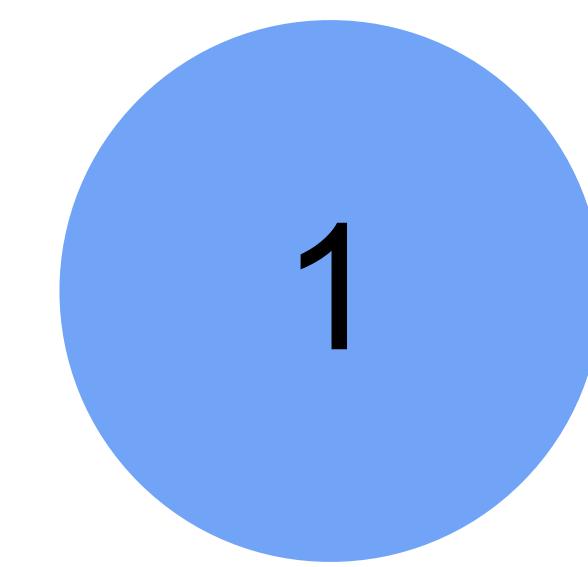
Sortieralgorithmen

Merge Sort



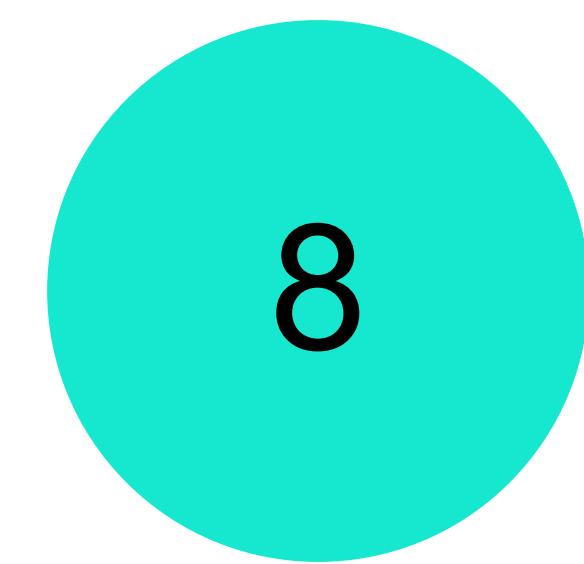
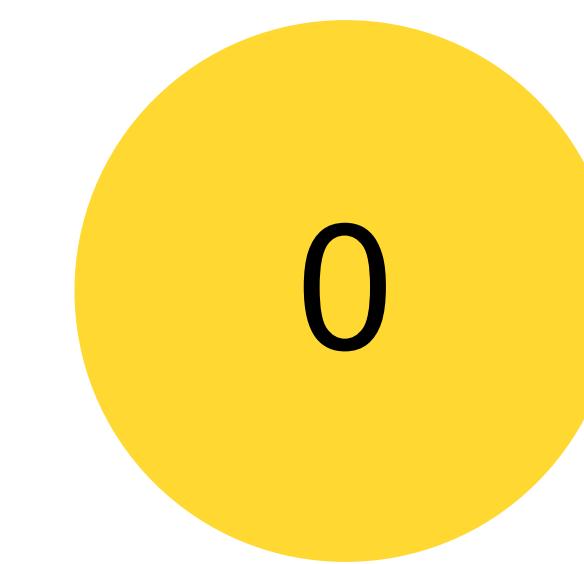
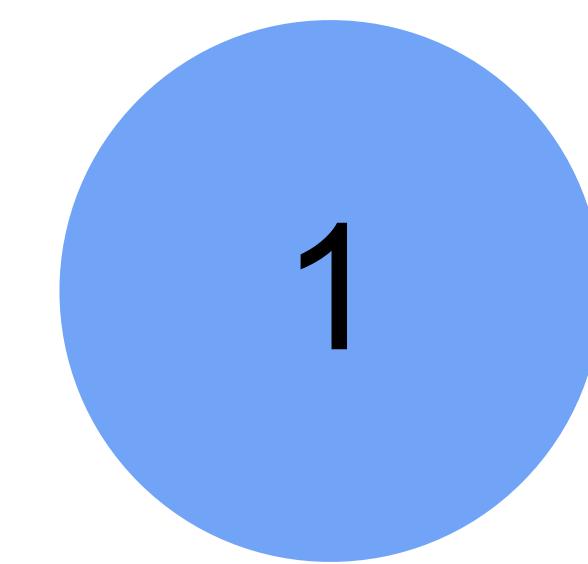
Sortieralgorithmen

Merge Sort



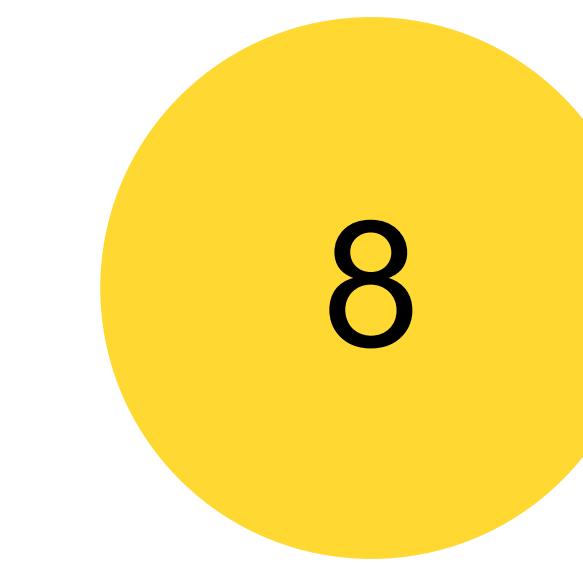
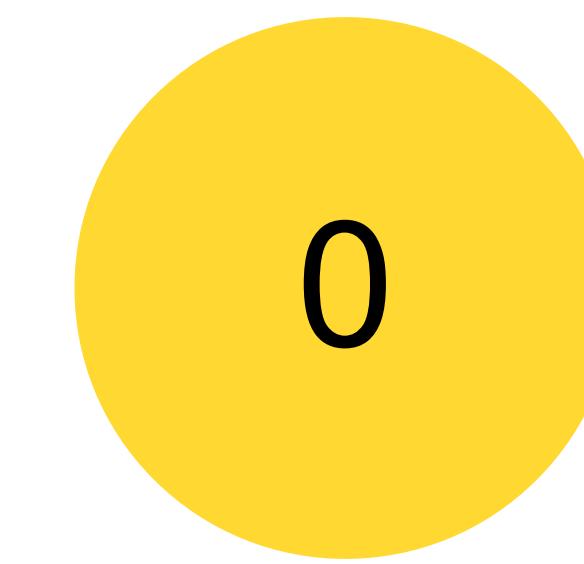
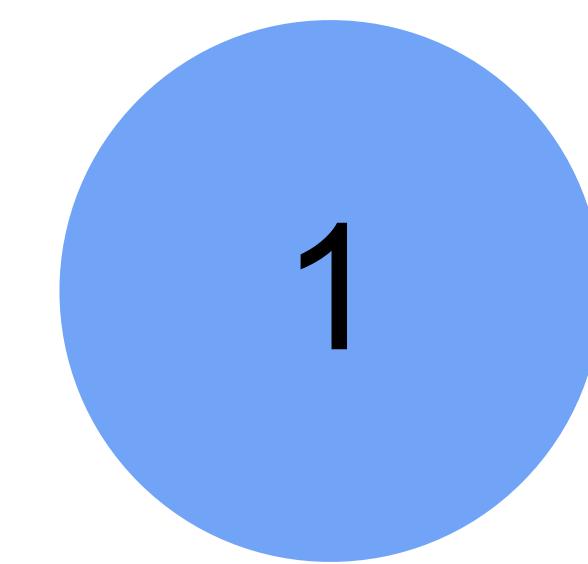
Sortieralgorithmen

Merge Sort



Sortieralgorithmen

Merge Sort



Sortieralgorithmen

Merge Sort

0

1

4

5

8

9

Sortieralgorithmen

Merge Sort

0

1

4

5

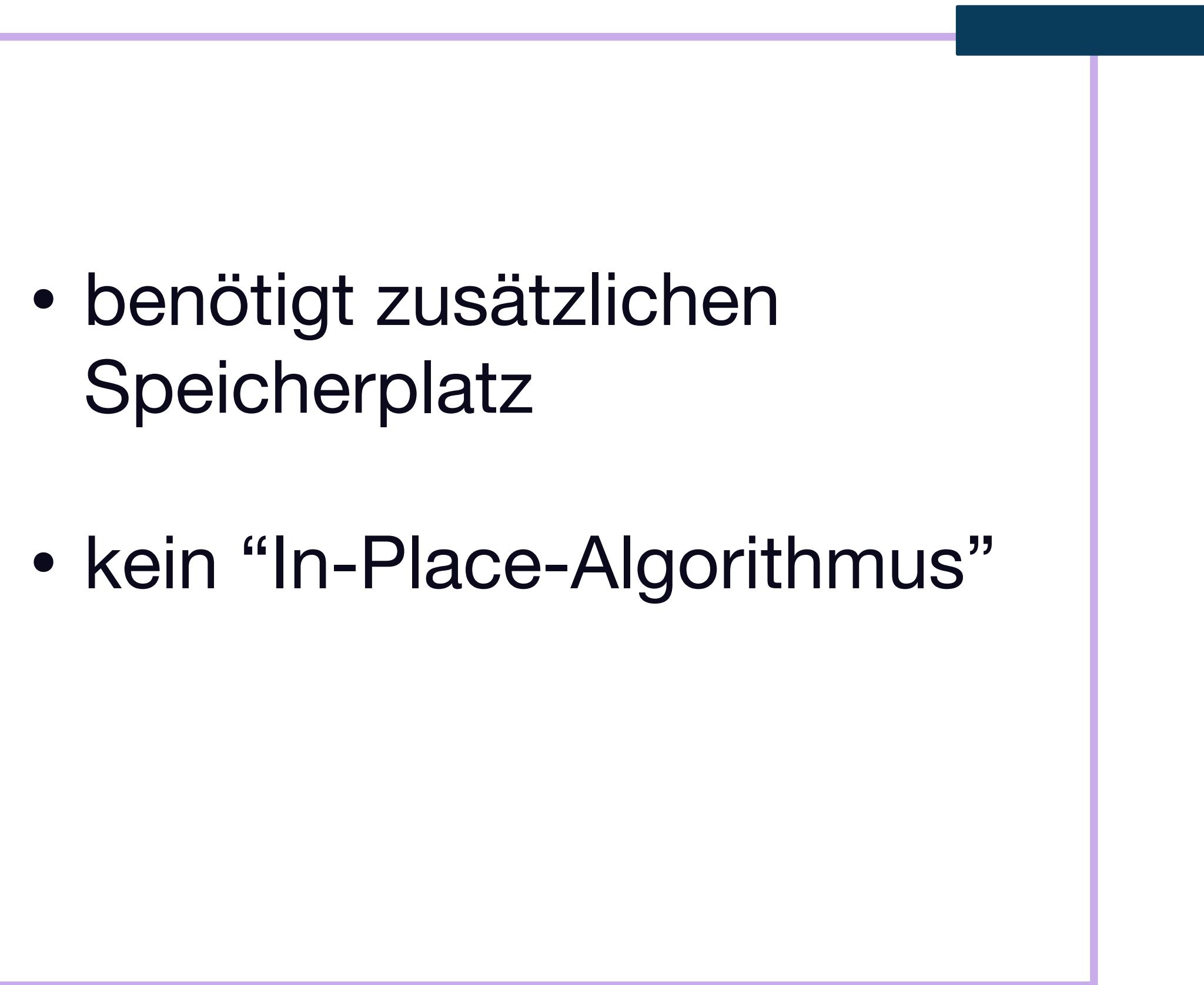
8

9

Vorteile und Nachteile



- Effizient: $O(n \log n)$ sowohl im Best-Case als auch im Worst-Case
- Einfache Implementierung



- benötigt zusätzlichen Speicherplatz
- kein “In-Place-Algorithmus”



Spiel: Sortieralgorithmen