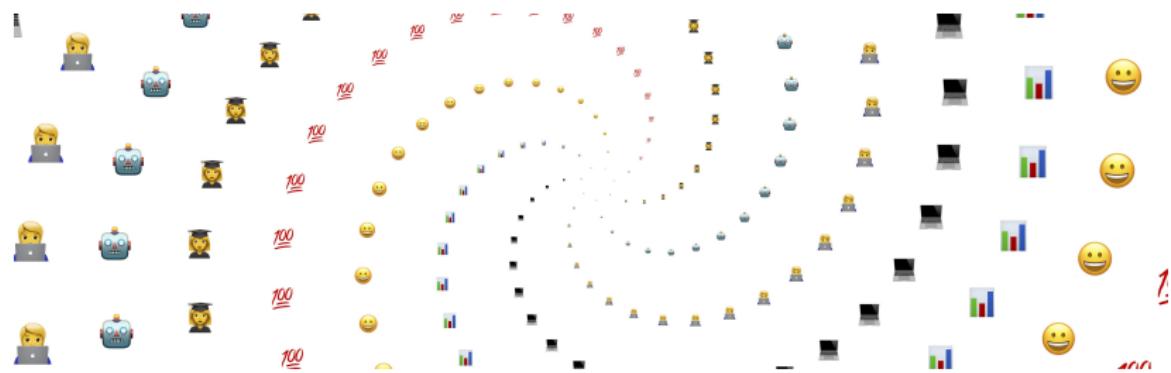


# Programmieren: Bubble Sort (Datenstrukturen)

## Listen

Cyril Wendl

Fachschaft Informatik  
Kantonsschule im Lee



# Einen Wert Suchen



Kantonsschule im Lee

# Einen Wert Suchen



Wie viele Dinge müssen wir im schlimmsten Fall anschauen, um einen Gegenstand im Koffer zu finden (wenn der Koffer  $n$  Gegenstände enthält)?

# Einen Wert Suchen



Kantonsschule im Lee

# Einen Wert Suchen



Und hier?



Kantonsschule im Lee

# Warum sind Sortier-Algorithmen wichtig?

- ▶ **Sortieren** ist eine der häufigsten Aufgaben in der Informatik.



# Warum sind Sortier-Algorithmen wichtig?

- ▶ **Sortieren** ist eine der häufigsten Aufgaben in der Informatik.
- ▶ Viele Anwendungen benötigen sortierte Daten:



# Warum sind Sortier-Algorithmen wichtig?

- ▶ **Sortieren** ist eine der häufigsten Aufgaben in der Informatik.
- ▶ Viele Anwendungen benötigen sortierte Daten:
  - ▶ Suchen in Telefonbüchern, Datenbanken, Online-Shops



# Warum sind Sortier-Algorithmen wichtig?

- ▶ **Sortieren** ist eine der häufigsten Aufgaben in der Informatik.
- ▶ Viele Anwendungen benötigen sortierte Daten:
  - ▶ Suchen in Telefonbüchern, Datenbanken, Online-Shops
  - ▶ Datenanalyse, Statistik, maschinelles Lernen



# Warum sind Sortier-Algorithmen wichtig?

- ▶ **Sortieren** ist eine der häufigsten Aufgaben in der Informatik.
- ▶ Viele Anwendungen benötigen sortierte Daten:
  - ▶ Suchen in Telefonbüchern, Datenbanken, Online-Shops
  - ▶ Datenanalyse, Statistik, maschinelles Lernen
  - ▶ Anzeige von Rankings, Highscores, Likes etc.



# Warum sind Sortier-Algorithmen wichtig?

- ▶ **Sortieren** ist eine der häufigsten Aufgaben in der Informatik.
- ▶ Viele Anwendungen benötigen sortierte Daten:
  - ▶ Suchen in Telefonbüchern, Datenbanken, Online-Shops
  - ▶ Datenanalyse, Statistik, maschinelles Lernen
  - ▶ Anzeige von Rankings, Highscores, Likes etc.
- ▶ **Zahlen:**



# Warum sind Sortier-Algorithmen wichtig?

- ▶ **Sortieren** ist eine der häufigsten Aufgaben in der Informatik.
- ▶ Viele Anwendungen benötigen sortierte Daten:
  - ▶ Suchen in Telefonbüchern, Datenbanken, Online-Shops
  - ▶ Datenanalyse, Statistik, maschinelles Lernen
  - ▶ Anzeige von Rankings, Highscores, Likes etc.
- ▶ **Zahlen:**
  - ▶ **Google** verarbeitet täglich über 14 Milliarden Suchanfragen (2025)



# Warum sind Sortier-Algorithmen wichtig?

- ▶ **Sortieren** ist eine der häufigsten Aufgaben in der Informatik.
- ▶ Viele Anwendungen benötigen sortierte Daten:
  - ▶ Suchen in Telefonbüchern, Datenbanken, Online-Shops
  - ▶ Datenanalyse, Statistik, maschinelles Lernen
  - ▶ Anzeige von Rankings, Highscores, Likes etc.
- ▶ **Zahlen:**
  - ▶ **Google** verarbeitet täglich über 14 Milliarden Suchanfragen (2025)
  - ▶ **Amazon** hat über 600 Millionen Produkte gelistet (2025)



# Warum sind Sortier-Algorithmen wichtig?

- ▶ **Sortieren** ist eine der häufigsten Aufgaben in der Informatik.
- ▶ Viele Anwendungen benötigen sortierte Daten:
  - ▶ Suchen in Telefonbüchern, Datenbanken, Online-Shops
  - ▶ Datenanalyse, Statistik, maschinelles Lernen
  - ▶ Anzeige von Rankings, Highscores, Likes etc.
- ▶ **Zahlen:**
  - ▶ **Google** verarbeitet täglich über 14 Milliarden Suchanfragen (2025)
  - ▶ **Amazon** hat über 600 Millionen Produkte gelistet (2025)
  - ▶ **Facebook** hat über 3.1 Milliarden aktive Nutzer (2025)



# Warum sind Sortier-Algorithmen wichtig?

- ▶ **Sortieren** ist eine der häufigsten Aufgaben in der Informatik.
- ▶ Viele Anwendungen benötigen sortierte Daten:
  - ▶ Suchen in Telefonbüchern, Datenbanken, Online-Shops
  - ▶ Datenanalyse, Statistik, maschinelles Lernen
  - ▶ Anzeige von Rankings, Highscores, Likes etc.
- ▶ **Zahlen:**
  - ▶ **Google** verarbeitet täglich über 14 Milliarden Suchanfragen (2025)
  - ▶ **Amazon** hat über 600 Millionen Produkte gelistet (2025)
  - ▶ **Facebook** hat über 3.1 Milliarden aktive Nutzer (2025)
- ▶ **Effizienz:**



# Warum sind Sortier-Algorithmen wichtig?

- ▶ **Sortieren** ist eine der häufigsten Aufgaben in der Informatik.
- ▶ Viele Anwendungen benötigen sortierte Daten:
  - ▶ Suchen in Telefonbüchern, Datenbanken, Online-Shops
  - ▶ Datenanalyse, Statistik, maschinelles Lernen
  - ▶ Anzeige von Rankings, Highscores, Likes etc.
- ▶ **Zahlen:**
  - ▶ **Google** verarbeitet täglich über 14 Milliarden Suchanfragen (2025)
  - ▶ **Amazon** hat über 600 Millionen Produkte gelistet (2025)
  - ▶ **Facebook** hat über 3.1 Milliarden aktive Nutzer (2025)
- ▶ **Effizienz:**
  - ▶ Google sortiert Milliarden Webseiten nach Relevanz



# Warum sind Sortier-Algorithmen wichtig?

- ▶ **Sortieren** ist eine der häufigsten Aufgaben in der Informatik.
- ▶ Viele Anwendungen benötigen sortierte Daten:
  - ▶ Suchen in Telefonbüchern, Datenbanken, Online-Shops
  - ▶ Datenanalyse, Statistik, maschinelles Lernen
  - ▶ Anzeige von Rankings, Highscores, Likes etc.
- ▶ **Zahlen:**
  - ▶ **Google** verarbeitet täglich über 14 Milliarden Suchanfragen (2025)
  - ▶ **Amazon** hat über 600 Millionen Produkte gelistet (2025)
  - ▶ **Facebook** hat über 3.1 Milliarden aktive Nutzer (2025)
- ▶ **Effizienz:**
  - ▶ Google sortiert Milliarden Webseiten nach Relevanz
  - ▶ Amazon sortiert Millionen Produkte nach Preis, Bewertung, etc.



# Warum sind Sortier-Algorithmen wichtig?

- ▶ **Sortieren** ist eine der häufigsten Aufgaben in der Informatik.
- ▶ Viele Anwendungen benötigen sortierte Daten:
  - ▶ Suchen in Telefonbüchern, Datenbanken, Online-Shops
  - ▶ Datenanalyse, Statistik, maschinelles Lernen
  - ▶ Anzeige von Rankings, Highscores, Likes etc.
- ▶ **Zahlen:**
  - ▶ **Google** verarbeitet täglich über 14 Milliarden Suchanfragen (2025)
  - ▶ **Amazon** hat über 600 Millionen Produkte gelistet (2025)
  - ▶ **Facebook** hat über 3.1 Milliarden aktive Nutzer (2025)
- ▶ **Effizienz:**
  - ▶ Google sortiert Milliarden Webseiten nach Relevanz
  - ▶ Amazon sortiert Millionen Produkte nach Preis, Bewertung, etc.
  - ▶ Facebook/Instagram sortieren Beiträge nach Aktualität oder Interesse



# Warum sind Sortier-Algorithmen wichtig?

- ▶ **Sortieren** ist eine der häufigsten Aufgaben in der Informatik.
- ▶ Viele Anwendungen benötigen sortierte Daten:
  - ▶ Suchen in Telefonbüchern, Datenbanken, Online-Shops
  - ▶ Datenanalyse, Statistik, maschinelles Lernen
  - ▶ Anzeige von Rankings, Highscores, Likes etc.
- ▶ **Zahlen:**
  - ▶ **Google** verarbeitet täglich über 14 Milliarden Suchanfragen (2025)
  - ▶ **Amazon** hat über 600 Millionen Produkte gelistet (2025)
  - ▶ **Facebook** hat über 3.1 Milliarden aktive Nutzer (2025)
- ▶ **Effizienz:**
  - ▶ Google sortiert Milliarden Webseiten nach Relevanz
  - ▶ Amazon sortiert Millionen Produkte nach Preis, Bewertung, etc.
  - ▶ Facebook/Instagram sortieren Beiträge nach Aktualität oder Interesse
- ▶ **Fazit:** Schnelle Sortieralgorithmen sparen Zeit, Geld und Rechenleistung!



# Sortier-Algorithmen

Wie kommen wir von  $x = [5, 3, 8, 20, 2, 10]$  zu  
 $x = [2, 3, 5, 8, 10, 20]$ ?

- ▶ **Algorithmus** = *eindeutige* und *endliche* Abfolge von Anweisungen oder Schritten zur Lösung eines Problems



# Sortier-Algorithmen

Wie kommen wir von  $x = [5, 3, 8, 20, 2, 10]$  zu  
 $x = [2, 3, 5, 8, 10, 20]$ ?

- ▶ **Algorithmus** = *eindeutige* und *endliche* Abfolge von Anweisungen oder Schritten zur Lösung eines Problems
- ▶ **Sortieren**: Wichtig, um Daten zu organisieren, Zugriff und die Suche nach Elementen effizienter zu machen und bildet die Grundlage für viele weitere Anwendungen und Algorithmen (z.B. *Suche*).



# Sortier-Algorithmen

Wie kommen wir von  $x = [5, 3, 8, 20, 2, 10]$  zu  
 $x = [2, 3, 5, 8, 10, 20]$ ?

- ▶ **Algorithmus** = eindeutige und endliche Abfolge von Anweisungen oder Schritten zur Lösung eines Problems
- ▶ **Sortieren**: Wichtig, um Daten zu organisieren, Zugriff und die Suche nach Elementen effizienter zu machen und bildet die Grundlage für viele weitere Anwendungen und Algorithmen (z.B. *Suche*).
- ▶ **Ziel**:



# Sortier-Algorithmen

Wie kommen wir von  $x = [5, 3, 8, 20, 2, 10]$  zu  
 $x = [2, 3, 5, 8, 10, 20]$ ?

- ▶ **Algorithmus** = eindeutige und endliche Abfolge von Anweisungen oder Schritten zur Lösung eines Problems
- ▶ **Sortieren**: Wichtig, um Daten zu organisieren, Zugriff und die Suche nach Elementen effizienter zu machen und bildet die Grundlage für viele weitere Anwendungen und Algorithmen (z.B. *Suche*).
- ▶ **Ziel**:

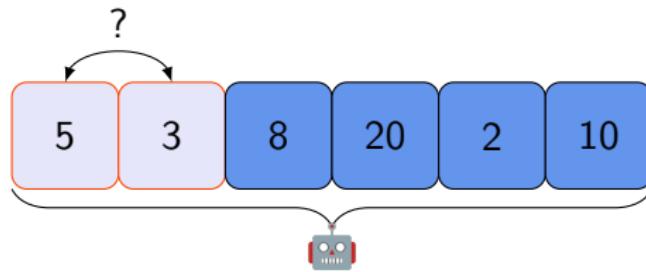


- ▶ Heute: **Bubble-Sort-Algorithmus**



# Bubble-Sort-Algorithmus

## Visualisierung



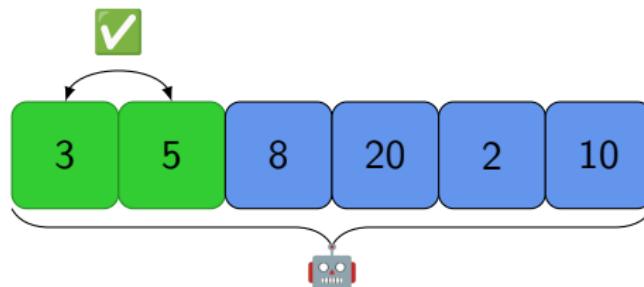
Schritt 1



Kantonsschule im Lee

# Bubble-Sort-Algorithmus

## Visualisierung

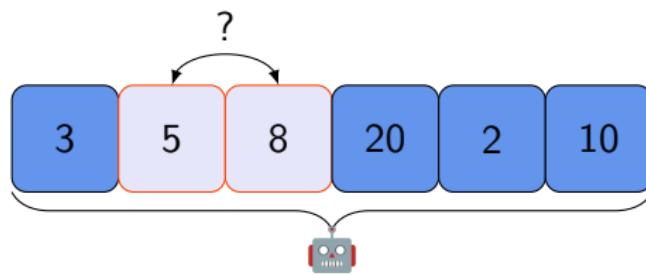


Schritt 1 (ausgetauscht!)



# Bubble-Sort-Algorithmus

## Visualisierung



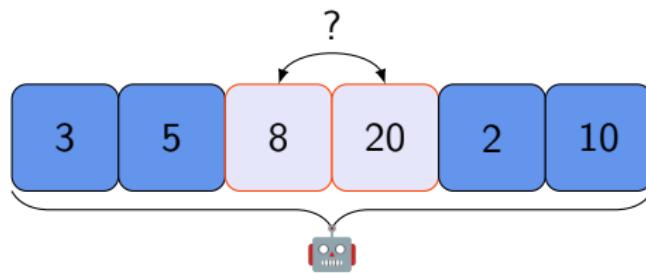
Schritt 2



Kantonsschule im Lee

# Bubble-Sort-Algorithmus

## Visualisierung



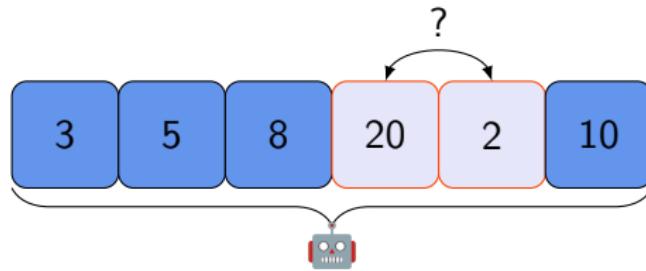
Schritt 3



Kantonsschule im Lee

# Bubble-Sort-Algorithmus

## Visualisierung



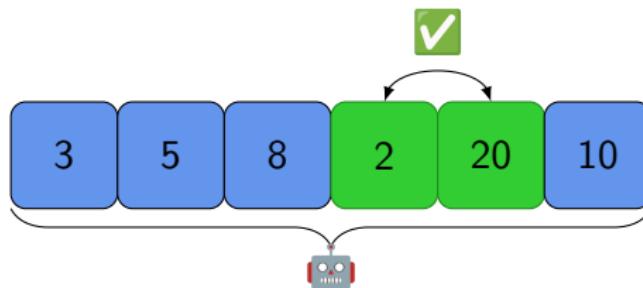
Schritt 4



Kantonsschule im Lee

# Bubble-Sort-Algorithmus

## Visualisierung

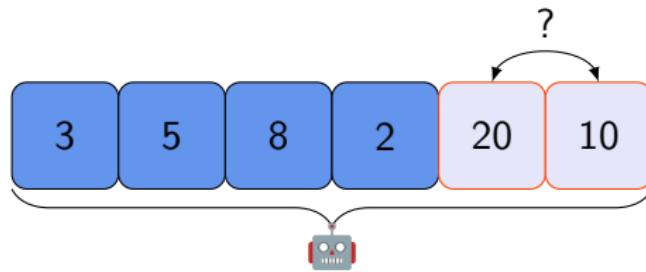


Schritt 4 (ausgetauscht!)



# Bubble-Sort-Algorithmus

## Visualisierung



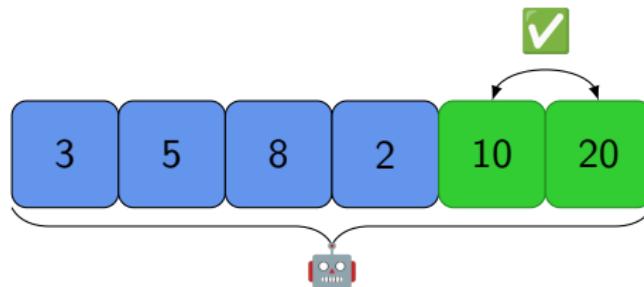
Schritt 5



Kantonsschule im Lee

# Bubble-Sort-Algorithmus

## Visualisierung

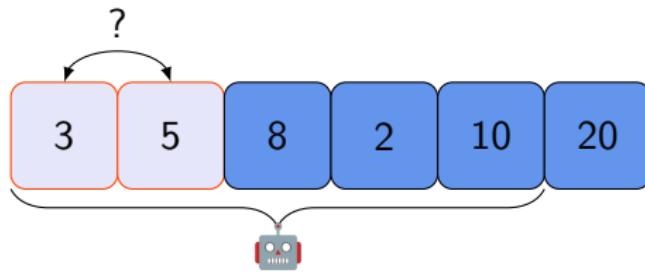


Schritt 5 (ausgetauscht!)



# Bubble-Sort-Algorithmus

## Visualisierung



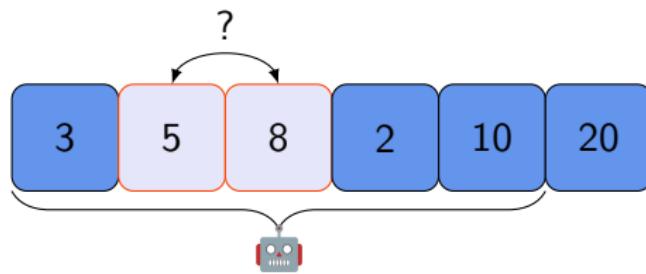
Schritt 6



Kantonsschule im Lee

# Bubble-Sort-Algorithmus

## Visualisierung



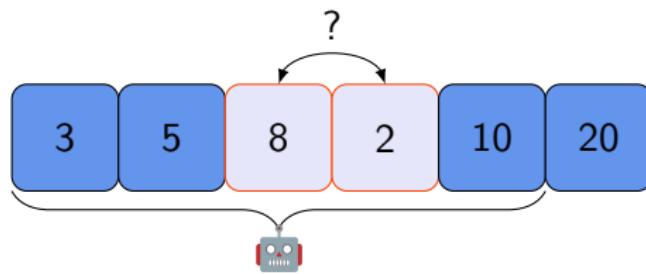
Schritt 7



Kantonsschule im Lee

# Bubble-Sort-Algorithmus

## Visualisierung



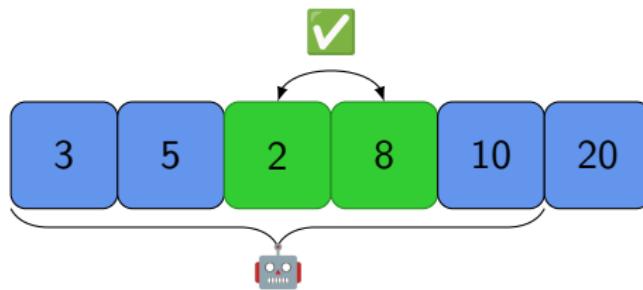
Schritt 8



Kantonsschule im Lee

# Bubble-Sort-Algorithmus

## Visualisierung

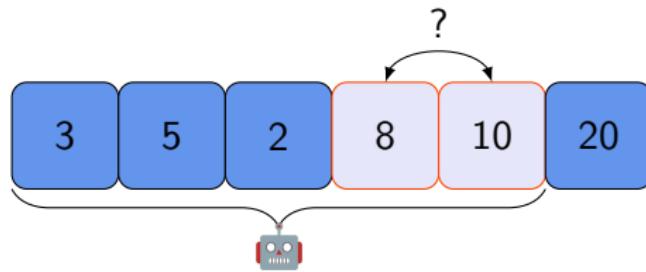


Schritt 8 (ausgetauscht!)



# Bubble-Sort-Algorithmus

## Visualisierung



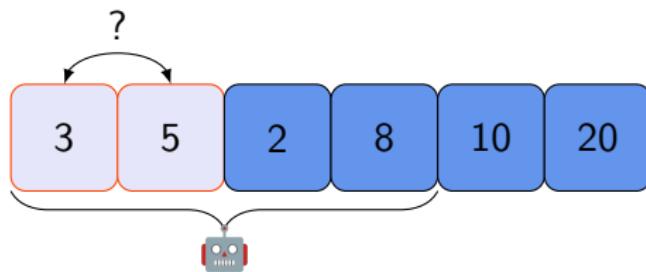
Schritt 9



Kantonsschule im Lee

# Bubble-Sort-Algorithmus

## Visualisierung



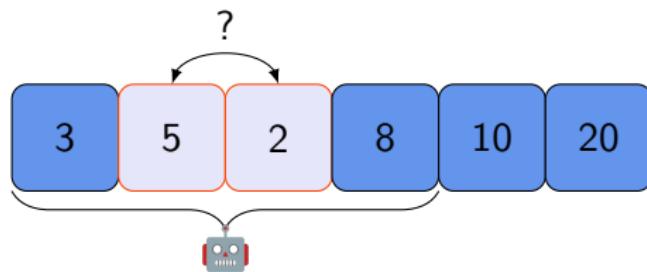
Schritt 10



Kantonsschule im Lee

# Bubble-Sort-Algorithmus

## Visualisierung



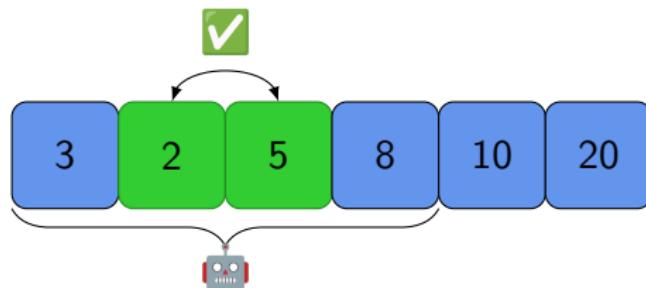
Schritt 11



Kantonsschule im Lee

# Bubble-Sort-Algorithmus

## Visualisierung

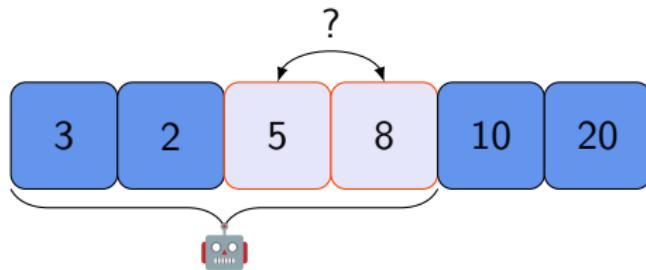


Schritt 11 (ausgetauscht!)



# Bubble-Sort-Algorithmus

## Visualisierung



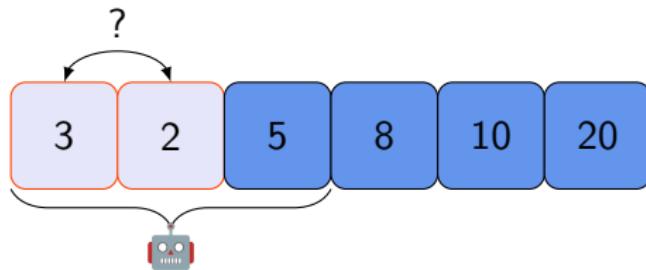
Schritt 12



Kantonsschule im Lee

# Bubble-Sort-Algorithmus

## Visualisierung



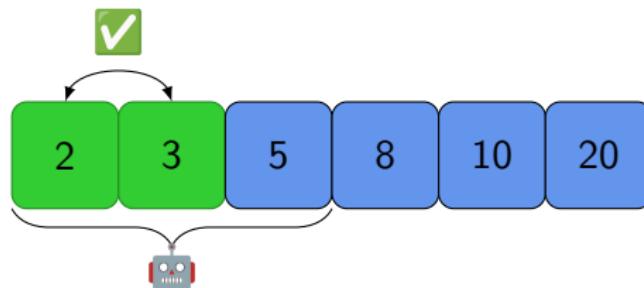
Schritt 13



Kantonsschule im Lee

# Bubble-Sort-Algorithmus

## Visualisierung

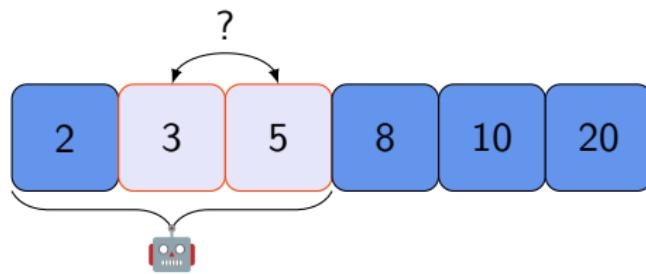


Schritt 13 (ausgetauscht!)



# Bubble-Sort-Algorithmus

## Visualisierung



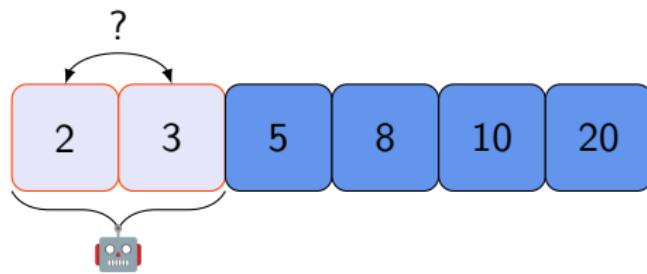
Schritt 14



Kantonsschule im Lee

# Bubble-Sort-Algorithmus

## Visualisierung



Schritt 15



Kantonsschule im Lee

# Bubble-Sort-Algorithmus

## Python-Code

Bevor wir uns den Algorithmus anschauen... Kann man Listen-Elemente so vertauschen?

```
1 # x ist die Liste, die sortiert werden soll
2 x = [5, 3, 8, 20, 2, 10]
3
4 x[0] = x[1]
5 print(x) # Was wird gedruckt?
6
7 x[1] = x[0]
8 print(x) # Was wird gedruckt?
```



# Bubble-Sort-Algorithmus

## Python-Code

Bevor wir uns den Algorithmus anschauen... Kann man Listen-Elemente so vertauschen?

```
1 # x ist die Liste, die sortiert werden soll
2 x = [5, 3, 8, 20, 2, 10]
3
4 x[0] = x[1]
5 print(x) # [3, 3, 8, 20, 2, 10]
6
7 x[1] = x[0]
8 print(x) # Was wird gedruckt?
```



# Bubble-Sort-Algorithmus

## Python-Code

Bevor wir uns den Algorithmus anschauen... Kann man Listen-Elemente so vertauschen?

```
1 # x ist die Liste, die sortiert werden soll
2 x = [5, 3, 8, 20, 2, 10]
3
4 x[0] = x[1]
5 print(x) # [3, 3, 8, 20, 2, 10]
6
7 x[1] = x[0]
8 print(x) # [3, 3, 8, 20, 2, 10]
```



# Bubble-Sort-Algorithmus

## Python-Code

Wir brauchen einen Zwischenspeicher!

```
# x ist die Liste, die sortiert werden soll
x = [5, 3, 8, 20, 2, 10]

# Zwischenspeicher-Variable a erstellen *bevor* wir x[0]
# überschreiben
a = x[0]

# Das erste Element von x wird überschrieben durch den zweiten
# Wert von x (wir verlieren also den ersten Wert, haben ihn
# aber noch in a als "Backup" in a gespeichert)
x[0] = x[1]
print(x)

# Jetzt können wir das "Backup" in das zweite Element von x
# einfügen
x[1] = a
print(x)
```

# Bubble-Sort-Algorithmus: Python-Code

```
# x ist die Liste, die sortiert werden soll
x = [5, 3, 8, 20, 2, 10]

for _ in range(len(x) - 1):
    i = 0
    for _ in range(len(x) - 1):
        # falls falsche Reihenfolge, dann...
        if x[i] > x[i + 1]:
            # ...Tausch von x[i] und x[i + 1]
            z = x[i]
            x[i] = x[i + 1]
            x[i + 1] = z

    i += 1
print(x)
```



# Bubble-Sort-Algorithmus: Python-Code (optimiert)

```
# x ist die Liste, die sortiert werden soll
x = [5, 3, 8, 20, 2, 10]

i = 0
for _ in range(len(x) - 1):
    j = 0
    for _ in range(len(x) - 1 - i):
        # falls falsche Reihenfolge, dann...
        if x[j] > x[j + 1]:
            # ...Tausch von x[j] und x[j + 1]
            z = x[j]
            x[j] = x[j + 1]
            x[j + 1] = z

        j += 1
print(x)
i += 1
```

# Laufzeit des Algorithmus

Anzahl Vergleiche, falls die Liste  $n$  lang ist



# Laufzeit des Algorithmus

Anzahl Vergleiche, falls die Liste  $n$  lang ist

- ▶ Nicht optimierte Version:

$$(n - 1)^2$$



# Laufzeit des Algorithmus

Anzahl Vergleiche, falls die Liste  $n$  lang ist

- ▶ Nicht optimierte Version:

$$(n - 1)^2$$

⇒ Falls die Liste 6 Elemente enthält:  $5^2 = 25$  Vergleiche



# Laufzeit des Algorithmus

Anzahl Vergleiche, falls die Liste  $n$  lang ist

- ▶ Nicht optimierte Version:

$$(n - 1)^2$$

⇒ Falls die Liste 6 Elemente enthält:  $5^2 = 25$  Vergleiche

- ▶ Optimierte Version:

$$\sum_{m=1}^{n-1} m$$



# Laufzeit des Algorithmus

Anzahl Vergleiche, falls die Liste  $n$  lang ist

- ▶ Nicht optimierte Version:

$$(n - 1)^2$$

⇒ Falls die Liste 6 Elemente enthält:  $5^2 = 25$  Vergleiche

- ▶ Optimierte Version:

$$\sum_{m=1}^{n-1} m$$

⇒ Falls die Liste 6 Elemente enthält:

$$\sum_{m=1}^5 m = 1 + 2 + 3 + 4 + 5 = 15 \text{ Vergleiche}$$



# Auftrag

- ▶ Moodle-Aufgaben „Bubble Sort“
- ▶  Aufgabe 6.12

