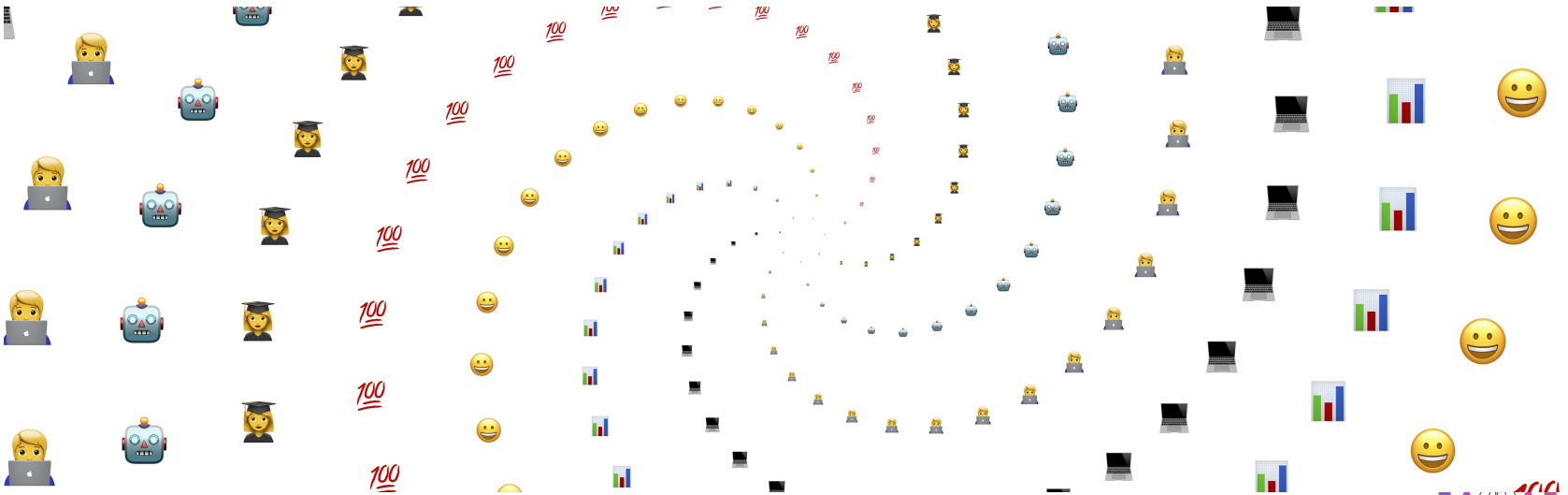
Programmieren: Binäre Suche (Datenstrukturen)

Listen

Cyril Blum

Fachschaft Informatik
Kantonsschule im Lee



Such-Algorithmen

- Wie finden wir schnellstmöglich heraus, ob eine bestimmte Zahl in einer *unsortierten* Liste enthalten ist? Z.B., ist die Zahl 3 in der Liste `x = [5, 3, 8, 20, 2, 10]`?
  $\rightarrow$ Wir können nichts anderes machen, als jedes Element durchschauen...

$\rightarrow$

Such-Algorithmen

- Wie finden wir schnellstmöglich heraus, ob eine bestimmte Zahl in einer *unsortierten* Liste enthalten ist? Z.B., ist die Zahl 3 in der Liste `x = [5, 3, 8, 20, 2, 10]`?
  $\rightarrow$ Wir können nichts anderes machen, als jedes Element durchschauen...


$\rightarrow$

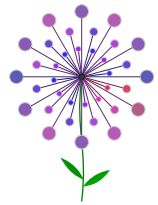

- Wie finden wir schnellstmöglich heraus, ob eine bestimmte Zahl in einer *sortierten* Liste enthalten ist? Z.B., ist die Zahl 3 in der Liste `x = [2, 3, 5, 8, 10, 20]`?

Such-Algorithmen

- Wie finden wir schnellstmöglich heraus, ob eine bestimmte Zahl in einer *unsortierten* Liste enthalten ist? Z.B., ist die Zahl 3 in der Liste `x = [5, 3, 8, 20, 2, 10]`?
  $\rightarrow$ Wir können nichts anderes machen, als jedes Element durchschauen...


$\rightarrow$


- Wie finden wir schnellstmöglich heraus, ob eine bestimmte Zahl in einer *sortierten* Liste enthalten ist? Z.B., ist die Zahl 3 in der Liste `x = [2, 3, 5, 8, 10, 20]`?
- **Suchen**: Vorgang, um ein Element in einer Menge von Daten *effizient* zu finden.

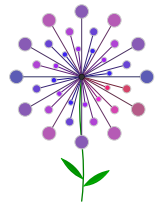


Such-Algorithmen

- Wie finden wir schnellstmöglich heraus, ob eine bestimmte Zahl in einer *unsortierten* Liste enthalten ist? Z.B., ist die Zahl 3 in der Liste `x = [5, 3, 8, 20, 2, 10]`?
  $\rightarrow$ Wir können nichts anderes machen, als jedes Element durchschauen...


$\rightarrow$

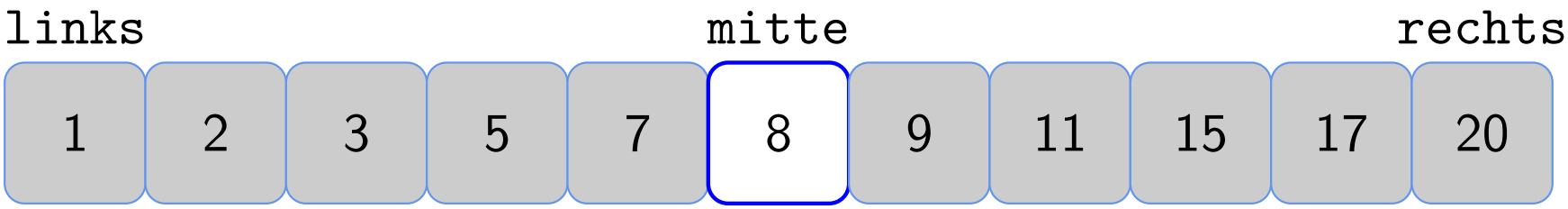

- Wie finden wir schnellstmöglich heraus, ob eine bestimmte Zahl in einer *sortierten* Liste enthalten ist? Z.B., ist die Zahl 3 in der Liste `x = [2, 3, 5, 8, 10, 20]`?
- **Suchen**: Vorgang, um ein Element in einer Menge von Daten *effizient* zu finden.
- Heute: **Binäre Suche**



Binäre Suche

Visualisierung

Gesuchte Zahl: 15

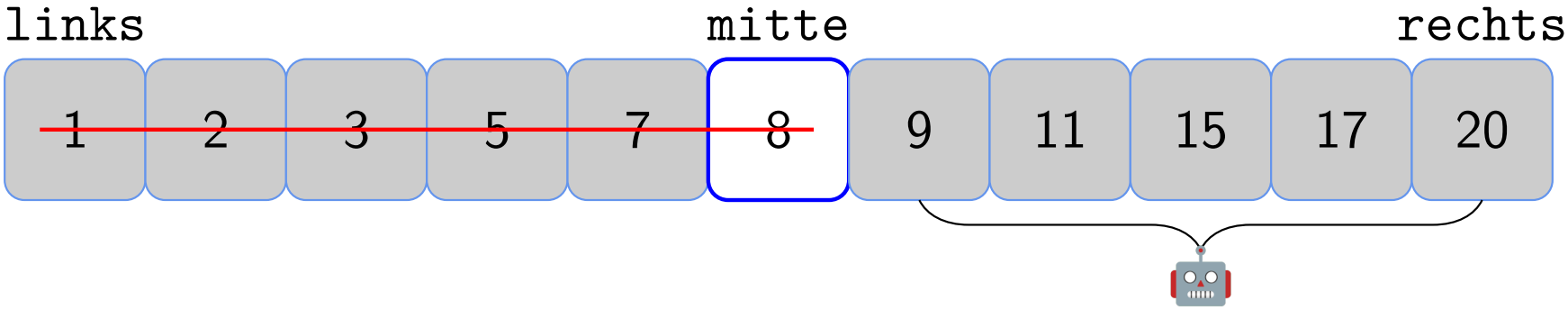


Schritt 1

Binäre Suche

Visualisierung

Gesuchte Zahl: 15

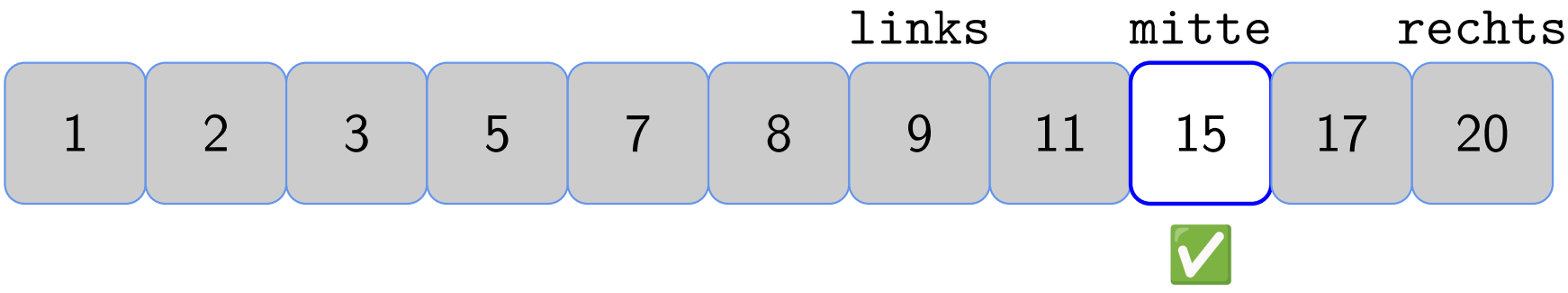


Schritt 1

Binäre Suche

Visualisierung

Gesuchte Zahl: 15



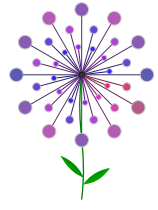
Schritt 2

Binäre Suche

```python
def binaere_suche(liste, ziel):
    # Definiere die Start- und Endpunkte des Suchbereichs
    links = 0
    rechts = len(liste) - 1

    # Solange der Suchbereich gültig ist, also links <= rechts
    while links <= rechts:
        # Berechne das mittlere Element
        mitte = (links + rechts) // 2

        # Wenn das mittlere Element das gesuchte Ziel ist, gib den Index zurück
        if liste[mitte] == ziel:
            print("Ziel Gefunden an Position", mitte)
            break

        # Wenn das Ziel grösser ist als das mittlere Element,
        # dann ist das Ziel im rechten Teil der Liste
        elif liste[mitte] < ziel:
            links = mitte + 1

        # Wenn das Ziel kleiner ist als das mittlere Element,
        # dann ist das Ziel im linken Teil der Liste
        else:
            rechts = mitte - 1


# Beispiel-Liste (muss sortiert sein)
meine_liste = [2, 3, 4, 10, 40]
ziel = 10

# Binäre Suche aufrufen
binaere_suche(meine_liste, ziel)
```

Auftrag

- Moodle-Aufgaben „Kapitel 6: Binäre Suche“
- Lesen Sie Beispiel 6.8
- Aufgabe 6.12-6.13



FreeFlower