

---

# 1 Einführung

---

## 1.1 Probleme in der Informatik

---

Ein **Problem** im Sinne der Informatik:

- Enthält Beschreibung der Eingabe
- Enthält Beschreibung der Ausgabe
- Gibt selbst **keinen** Übergang von Ein und Ausgabe an



ABBILDUNG 1: Modell Problem Informatik

z.B. Finde den kürzesten Weg zwischen 2 Orten

Eine **Probleminstanz** ist eine konkrete Eingabebelegung für die entsprechende Ausgabe gewünscht.

Für das obige Problem wäre das z.B. „Was ist der kürzeste Weg vom Audimax in die Mensa?“

---

## 1.2 Definitionen für Algorithmen

---

„Ein Algorithmus ist eine **endliche Folge** von Rechenschritten, die eine **Eingabe** in eine **Ausgabe** umwandelt.“<sup>1</sup>

**Anforderungen an Algorithmen:**

- Spezifizierung der Ein- und Ausgabe:
  - Anzahl und Typen aller Elemente ist/sind definiert
- Eindeutigkeit:
  - Jeder Einzelschritt ist klar definiert und ausführbar
  - Die Reihenfolge der Einzelschritte ist festgelegt.
- Endlichkeit
  - Notation hat endliche Länge

**Eigenschaften von Algorithmen:**

- Determiniertheit:
  - Für gleiche Eingabe folgt stets die gleiche Ausgabe (andere Zwischenzustände sind möglich)
- Determinismus:
  - Für die gleiche Eingabe ist die Ausführung und Ausgabe stets identisch.
- Terminierung:
  - Der Algorithmus läuft für jede endliche Eingabe nur endlich lange
- Korrektheit:
  - Der Algorithmus berechnet stets die spezifizierte Ausgabe (falls dieser terminiert).
- Effizienz:
  - Sparsamkeit im Ressourcenverbrauch (Zeit, Speicher, Energie, ...)

---

<sup>1</sup>Quelle: Cormen et al., 4. Auflage

---

### 1.3 Definitionen für Datenstrukturen

---

„Eine Datenstruktur ist eine Methode, Daten **abzuspeichern** und zu **organisieren** sowie den **Zugriff** auf die Daten und die **Modifikation** der Daten zu erleichtern.“<sup>2</sup>

Datenstrukturen:

- Sind Organisationsformen für Daten
- Beinhalten Strukturbestandteile und Nutzerdaten (Payload)

z.B. Arrays, Listen, ...

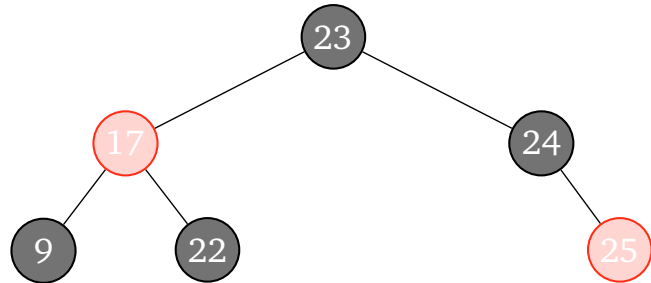


ABBILDUNG 2: Beispiel Datenstruktur (Rot-Schwarz-Baum)

---

### 1.4 Pseudocode-Konventionen

---

- Blöcke werden durch Einrückungen hervorgehoben
- Blockkonstrukte sind „for x to y“, „while“, „repeat-until“ und „if-else“
- Kommentare erhalten das Prädikat „/ /“
- „i = j = e“ bedeutet, i und j erhalten den Wert von e
- Variablen sind immer lokal
- $A[i]$  bezeichnet das i-te Element im Array A
- $A[i..j]$  Array A im Bereich von  $i - j$
- Attribute werden über einen „.“ abgerufen

---

### 1.5 Weitere wichtige Definitionen

---

- **short circuit evaluation (Kurzschlussauswertung)**: Strategie, bei der die Auswertung, nachdem die Gesamtlösung durch einen Teilausdruck eindeutig bestimmt wurde, abgebrochen wird. z.B.:  $1+1==2 \ || \ 1/0==0 \rightarrow \text{true}$
- **call-by-reference (Referenzparameter)**: Übergeben von Referenz auf ein Objekt. Dadurch sind Änderungen an diesem innerhalb der Routine möglich.
- **call-by-value (Wertparameter)**: Übergeben einer Kopie des Objekts. Das ursprüngliche Objekt kann so nicht mehr verändert werden, jedoch die Kopie. Die referenzen bleiben beim Kopieren gleich (z.B. beim Linked List)

---

<sup>2</sup>Quelle: Cormen et al., 4. Auflage