

# Informatik E2 Abels



Algorithmus

# Algorithmus

Ein **Algorithmus** ist eine eindeutige Handlungsvorschrift zur Lösung eines Problems oder einer Klasse von Problemen.

Algorithmen bestehen aus endlich vielen, wohldefinierten Einzelschritten.

Somit können sie zur Ausführung in einem Computerprogramm implementiert, aber auch in menschlicher Sprache formuliert werden.

Bei der Problemlösung wird eine bestimmte Eingabe in eine bestimmte Ausgabe überführt.

Beispiel: Kochrezept

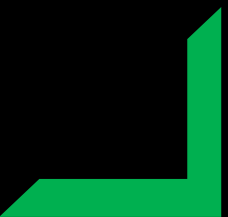




# Übung 1

Finde zwei Beispiele aus deinem Alltag für Algorithmen, die ...

- ... eindeutig, endlich und implementierbar sind.
- ... nicht eindeutig sind.
- ... nicht endlich sind.
- ... nicht implementierbar sind.



# Eigenschaften eines Algorithmus

# Eigenschaften eines Algorithmus

## Endlichkeit

Ein Algorithmus besteht aus endlich vielen Anweisungen (Verarbeitungsbefehlen bzw. Regeln) endlicher Länge.

# Eigenschaften eines Algorithmus

Endlichkeit

Eindeutigkeit

deterministisch

Mit jeder Anweisung ist auch die nächstfolgende festgelegt.

determiniert

Gleiche Eingaben führen bei wiederholter Ausführung zu den gleichen Ausgaben.

# Eigenschaften eines Algorithmus

Endlichkeit

Eindeutigkeit

Ausführbarkeit

Jede einzelne Anweisung muss für den Ausführenden des Algorithmus (den „Prozessor“) verständlich und ausführbar sein.

Ein Algorithmus ist also immer nur ein Algorithmus bezüglich eines Prozessors.



# Eigenschaften eines Algorithmus

Endlichkeit

Eindeutigkeit

Ausführbarkeit

Allgemeingültigkeit

Ein Algorithmus muss auf alle Aufgaben gleichen Typs (Aufgabenklasse) anwendbar sein und stets zum gesuchten Resultat (Lösung) führen.

# Eigenschaften eines Algorithmus

Endlichkeit

+

Eindeutigkeit

+

Ausführbarkeit

+

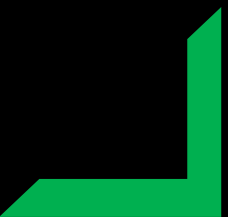
Allgemeingültigkeit



## Übung 2

Entscheide für die folgenden Probleme, ob für deren Lösung ein Algorithmus angegeben werden kann. Berücksichtige dabei die genannten Eigenschaften von Algorithmen.

1. Prüfen, ob eine Zahl durch 2 teilbar ist.
2. Lösen einer quadratischen Gleichung der Form  $ax^2 + bx + c = 0$
3. Auflisten aller geraden natürlichen Zahlen.
4. Schreiben von 15 Punkten in der nächsten Informatik-Klausur.
5. Schreiben eines Liebesbriefs.
6. Wechseln eines Autoreifens.
7. Konstruieren eines Kreises durch drei Punkte, die nicht auf einer Geraden liegen.



Beispiel:

Euklidischer Algorithmus

# Euklidischer Algorithmus

a →

b →

- Bei gleichen Zahlen ist a der ggT
- Teile die größere durch die kleinere Zahl
- Geht die Division auf, ist der Divisor der ggT
- Geht die Division nicht auf, bleibt ein Rest.  
Dieser ist der neue Divisor, der alte wird zum Dividenden.
- Setze das Verfahren fort, bis der Rest 1 (teilerfremd) oder 0 ist.

→ ggT(a,b)

# Euklidischer Algorithmus

544



391



- Bei gleichen Zahlen ist a der ggT
- Teile die größere durch die kleinere Zahl
- Geht die Division auf, ist der Divisor der ggT
- Geht die Division nicht auf, bleibt ein Rest. Dieser ist der neue Divisor, der alte wird zum Dividenden.
- Setze das Verfahren fort, bis der Rest 1 (teilerfremd) oder 0 ist.



17

$$544 : 391 = 1 \text{ R } 153$$

$$391 : 153 = 2 \text{ R } 85$$

$$153 : 85 = 1 \text{ R } 68$$

$$85 : 68 = 1 \text{ R } 17$$

$$68 : 17 = 4 \text{ R } 0$$

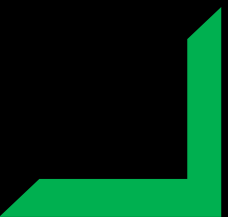


## Übung 3

Bestimme mithilfe des Euklidischen Algorithmus den ggT von ...

- ... 420 und 1001.
- ... 11 und 11.
- ... 3 und 22.

Fällt dir auf, welche Beispiele ich bewusst gewählt habe, um den Algorithmus zu testen?

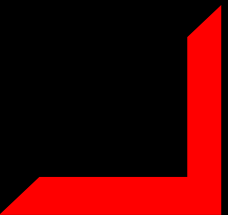




# Tagebucheintrag

Algorithmus

Euklidischer Algorithmus







# Wochenübung

Führe das Programm **EuklidischerAlgorithmus.py** aus und teste es mit allen relevanten Testfällen.

```
EuklidischerAlgorithmus.py

# Eingabe
n1 = int(input("a: "))
n2 = int(input("b: "))

# Verarbeitung
a = n1
b = n2
r = a % b
while r:
    a = b
    b = r
    r = a % b

# Ausgabe
print("Der ggT von " + str(n1) + " und " + str(n2) + " ist: " + str(b))
```

