Der Algorithmus von Euklid   Lösungen   Theorie

Einleitung

Der grösste gemeinsame Teiler (ggT) von zwei ganzen Zahlen a und b ist die grösste natürliche Zahl, die sowohl a als auch b teilt.

Beispiele:

(a) $\mathrm{ggT}(21, 15) = 3$

(b) $\mathrm{ggT}(3, 17) = 1$

(c) $\mathrm{ggT}(31, 31) = 31$

Spezialfälle:

(d) $\mathrm{ggT}(14, 0) = 14$

(e) $\mathrm{ggT}(-6, -8) = \mathrm{ggT}(6, 8) = 2$

(f) $\mathrm{ggT}(0, 0) =$ nicht definiert

Lösungsidee 1

Ist genau eine der beiden Zahlen null, dann gilt $\mathrm{ggT}(a, 0) = a$.

Andernfalls gilt $a \geq b > 0$. Dividiere die beiden Zahlen jeweils durch b, $b-1$, $b-2$, ..., 3, 2, 1. Die erste dieser Zahlen, die a *und* b ohne Rest teilt, muss der grösste gemeinsame Teiler sein.

Beispiel: $\mathrm{ggT}(12, 8)$

$$\begin{array}{lcl}
12 \bmod 8 \neq 0 & \text{und} & 8 \bmod 8 = 0 \\
12 \bmod 7 \neq 0 & \text{und} & 8 \bmod 7 \neq 0 \\
12 \bmod 6 = 0 & \text{und} & 8 \bmod 6 \neq 0 \\
12 \bmod 5 \neq 0 & \text{und} & 8 \bmod 5 \neq 0 \\
12 \bmod 4 = 0 & \text{und} & 8 \bmod 4 = 0
\end{array}$$

$\Rightarrow \quad \mathrm{ggT}(12, 8) = 4$

Lösungsidee 2

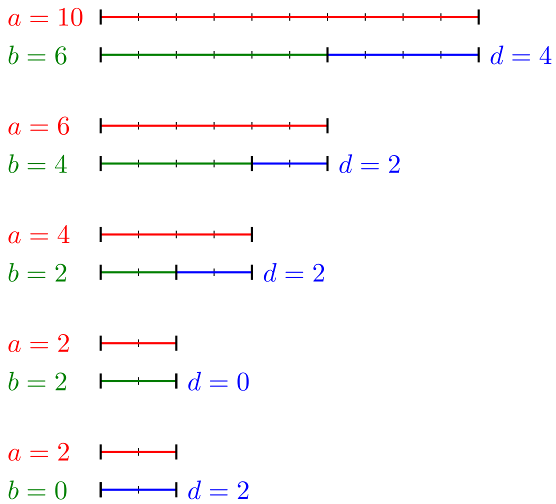
Wenn eine Zahl t, die Zahlen a und b (mit $a \geq b$) teilt, dann teilt t auch ihre Differenz $d = a - b$.

Beispiel:

$\mathrm{ggT}(10, 6) = \mathrm{ggT}(6, 4) = \mathrm{ggT}(4, 2) = \mathrm{ggT}(2, 2) = \mathrm{ggT}(2, 0) = 2$

Idee: Euklid von Alexandria (griechischer Mathematiker, ca. 3. Jh. v. Chr.)

Der Versuch einer Visualisierung



$\mathrm{ggT}(10, 6) = \mathrm{ggT}(2, 0) = 2$

Der klassische Algorithmus von Euklid

```
EUCLID_CLASSIC(a, b)
  while b != 0
    if a < b:
      vertausche a und b
    end if
    a - b -> a
  end while
  return b
```

Ein *Algorithmus* ist eine in der Beschreibung und Ausführung endliche und eindeutige Vorschrift zur Lösung einer Klasse von Problemen.

Aufgabe 1

a	b	b != 0
21	15	true
6	15	true
15	6	true
9	6	true
3	6	true
6	3	true
3	3	true
0	3	true
3	0	false $\Rightarrow$ ggT(21,15)=3

Aufgabe 2

Implementiere den Algorithmus von Euklid als Python-Funktion `gcdClassic(a,b)`, die den ggT von a und b als Wert zurückgibt oder `None`, wenn der ggT nicht definiert ist.

(*gcd* steht für *greatest common divisor*)

Aufgabe 3

Berechne $\text{ggT}(12, 2)$. Was fällt dabei auf?

a	b	b! = 0
12	2	true
10	2	true
8	2	true
6	2	true
4	2	true
2	2	true
2	0	false ⇒ ggT(12, 2) = 2

Der Algorithmus subtrahiert ständig eine kleine Zahl und kommt so nur langsam ans Ziel.

Der Algorithmus von Euklid mit Divsionsrest

Um wiederholte Subtraktionen zu vermeiden, verwendet man heute den Algorithmus von Euklid mit der Modulo-Funktion, d. h. man berechnet den Divsionsrest anstelle der Division.

```
EUCLID_MOD(a, b)
  if a = 0
    return b
  end if
  while b != 0
    mod(a, b) -> r
    a = b
    b = r
  end while
  return a
```

Aufgabe 4

Berechne $\text{ggT}(12, 2)$ mit der Modulo-Version des euklidschen Algorithmus'.

a	b	b != 0	r = a % b
12	2	false	0
2	0	true	--

$\text{ggT}(12, 2) = 2$

Aufgabe 5

Berechne $\text{ggT}(48, 30)$ mit der Modulo-Version des euklidschen Algorithmus'.

a	b	b! = 0	r = a % b
48	30	false	18
30	18	false	12
18	12	false	6
12	6	false	0
6	0	true	--

$\text{ggT}(48, 30) = 6$

Aufgabe 6

Implementiere den Algorithmus von Euklid mit Divsionsrest als Python-Funktion mit dem Namen `gcdMod(a,b)`.