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) $\text{ggT}(21, 15) =$

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

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

Spezialfälle:

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

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

(f) $\text{ggT}(0, 0) =$

Lösungsidee 1

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

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: $\text{ggT}(12, 8)$

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

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

Lösungsidee 2

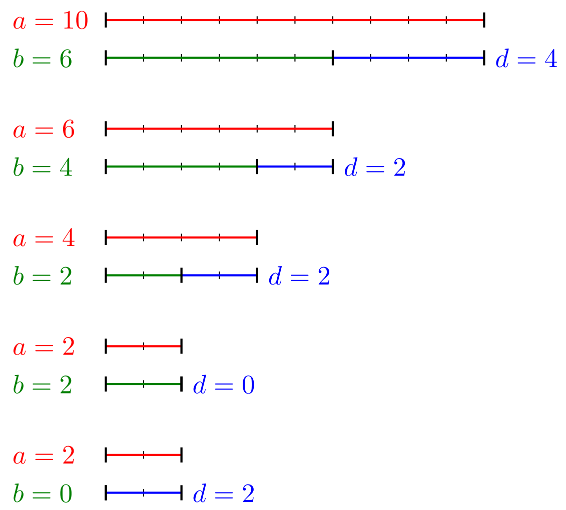
Wenn eine Zahl t, die Zahlen a und b (mit $a \geq b$) teilt, dann teilt t auch ihre Differenz $d = a - b$.

Beispiel:

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

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
```

Aufgabe 1

a	b	b != 0
21	15	

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	

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		

Aufgabe 5

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

a	b	b! = 0	r = a % b
48	30		

Aufgabe 6

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