GGT-Algorithmen

Übungen

Aufgabe 1

Gegeben: Primfaktorzerlerungen der natürlichen Zahlen a und b

$$71500 = 5 \cdot 2 \cdot 5 \cdot 11 \cdot 5 \cdot 13 \cdot 2$$
$$4200 = 2 \cdot 5 \cdot 2 \cdot 5 \cdot 7 \cdot 3 \cdot 2$$

Gesucht: $\mathrm{ggT}(71500, 4200) =$

Aufgabe 1

$$71500 = 5 \cdot 2 \cdot 5 \cdot 11 \cdot 5 \cdot 13 \cdot 2$$
$$4200 = 2 \cdot 5 \cdot 2 \cdot 5 \cdot 7 \cdot 3 \cdot 2$$
$$\mathrm{ggT}(71500, 4200) = 2 \cdot 2 \cdot 5 \cdot 5 = 100$$

Aufgabe 2

Wie lautet die korrekt ausgeschriebene englische Bezeichnung für den grössten gemeinsamen Teiler und die dazu gehörende Abkürzung?

Aufgabe 2

greatest common divisor (gcd)

Aufgabe 3

Zeige schrittweise, wie der klassische Algorithmus von Euklid den grössten gemeinsamen Teiler der natürlichen Zahlen $a = 80$ und $b = 32$ berechnet.

$\mathrm{ggT}(80, 32) =$

Aufgabe 3

$$\begin{aligned}
\mathrm{ggT}(80,32) &= \mathrm{ggT}(48,32) = \mathrm{ggT}(16,32) = \mathrm{ggT}(32,16)\\
&= \mathrm{ggT}(16,16) = \mathrm{ggT}(0,16) = \mathrm{ggT}(16,0)\\
&= \mathrm{ggT}(16,0) = 16
\end{aligned}$$

Aufgabe 4

Untersuche, was passiert, wenn du den klassischen euklidischen Algorithmus mit einem negativen ersten Operanden ausführst.

Führe konsequent vier bis fünf Schritte des Algorithmus aus. Beschreibe das Problem, das dabei ensteht und formuliere einen Vorschlag zu seiner Lösung.

$\text{ggT}(-8, 10) =$

Aufgabe 4

$$
\begin{aligned}
\operatorname{ggT}(-8, 10) &= \operatorname{ggT}(10, -8) = \operatorname{ggT}(18, -8) = \operatorname{ggT}(26, -8) \\
&= \operatorname{ggT}(34, -8) = \operatorname{ggT}(42, -8) = \dots
\end{aligned}
$$

Im ersten Schritt werden die beiden Operanden vertauscht. Danach wird der erste Operand immer grösser und der zweite verändert sich nicht mehr. Da er so nicht null werden kann, bricht das Verfahren nicht ab und läuft endlos weiter.

Das Problem kann gelöst werden, indem man vor der Ausführung des Algorithmus die Operanden a und b durch ihre Absolutbeträge $|a|$ und $|b|$ ersetzt.

Aufgabe 5

Beschreibe den wesentlichen Nachteil des klassischen euklidischen Algorithmus anhand eines gut gewählten Beispiels mit zwei positiven Operanden und formuliere einen Verbesserungsvorschlag.

Aufgabe 5

$$
\begin{aligned}
\mathrm{ggT}(100,3) &= \mathrm{ggT}(97,3) = \mathrm{ggT}(94,3) = \mathrm{ggT}(91,3) \\
&= \mathrm{ggT}(88,3) = \mathrm{ggT}(85,3) = \mathrm{ggT}(82,3) \\
&= \dots \\
&= \mathrm{ggT}(4,3) = \mathrm{ggT}(1,3) = \mathrm{ggT}(3,1) \\
&= \mathrm{ggT}(2,1) = \mathrm{ggT}(1,1) = \mathrm{ggT}(1,0) = 1
\end{aligned}
$$

Wenn der Unterschied zwischen den beiden Operanden sehr gross ist, verbringt der Algorithmus die meiste Zeit damit, eine kleine Zahl von einer grossen Zahl zu subtrahieren, bis der erste Operand kleiner als der zweite ist.

Das Problem kann gelöst werden, indem man statt der Subtraktion $a - b$ den Divsionsrest $r = a \bmod b$ berechnet und dann den ersten Operanden a durch den zweiten Operanden und den zweiten Operanden durch den Divisionsrest ersetzt. Danach ist der erste Operand immer grösser als der zweite, so dass keine Vertauschungen mehr nötig sind.

Aufgabe 6

Zeige schrittweise, wie der moderne Algorithmus von Euklid den grössten gemeinsamen Teiler der natürlichen Zahlen $a = 60$ und $b = 84$ berechnet.

$\mathrm{ggT}(60, 84) =$

Aufgabe 6

$$\begin{aligned}\operatorname{ggT}(60,84) &= \operatorname{ggT}(84,60) = \operatorname{ggT}(60,24) = \operatorname{ggT}(24,12)\\ &= \operatorname{ggT}(12,0) = 12\end{aligned}$$

Aufgabe 7

Zeige, schrittweise, wie die klassische und die moderne Version des Algorithmus von Euklid $\mathrm{ggT}(64, 44)$ berechnen und vergleiche den Aufwand.

Aufgabe 7

$$
\begin{aligned}
\text{klassisch: } \operatorname{ggT}(64,44) &= \operatorname{ggT}(20,44) = \operatorname{ggT}(44,20) = \operatorname{ggT}(24,20)\\
&= \operatorname{ggT}(4,20) = \operatorname{ggT}(20,4) = \operatorname{ggT}(16,4)\\
&= \operatorname{ggT}(12,4) = \operatorname{ggT}(8,4) = \operatorname{ggT}(4,4)\\
&= \operatorname{ggT}(0,4) = \operatorname{ggT}(4,0) = \operatorname{ggT}(4,0)\\
&= 4
\end{aligned}
$$

$$
\begin{aligned}
\text{modern: } \operatorname{ggT}(64,44) &= \operatorname{ggT}(44,20) = \operatorname{ggT}(20,4) = \operatorname{ggT}(4,0)\\
&= 4
\end{aligned}
$$

klassisch: 13 Schritte

modern: 4 Schritte

Aufgabe 8

Zeige, schrittweise, wie die klassische und die moderne Version des Algorithmus von Euklid ggT$(34, 21)$ berechnen. Vergleiche die Anzahl der Rechenschritte, wobei bei der klassischen Version nur die Subtraktionen und beim der modernen Version nur die die Berechnung der Divsionsreste zu zählen sind.

(a) Was stellst du fest? (b) Woran liegt das?

Aufgabe 8

klassisch:
$$\begin{aligned}\mathrm{ggT}(34,21) &= \mathrm{ggT}(13,21) = \mathrm{ggT}(21,13)\\ &= \mathrm{ggT}(8,13) = \mathrm{ggT}(13,8)\\ &= \mathrm{ggT}(5,8) = \mathrm{ggT}(8,5)\\ &= \mathrm{ggT}(3,5) = \mathrm{ggT}(5,3)\\ &= \mathrm{ggT}(2,3) = \mathrm{ggT}(3,2)\\ &= \mathrm{ggT}(1,2) = \mathrm{ggT}(2,1)\\ &= \mathrm{ggT}(1,1) = \mathrm{ggT}(0,1)\\ &= \mathrm{ggT}(1,0) = \mathrm{ggT}(1,0)\\ &= 1\end{aligned}$$

modern:
$$\begin{aligned}\mathrm{ggT}(34,21) &= \mathrm{ggT}(21,13) = \mathrm{ggT}(13,8) = \mathrm{ggT}(8,5)\\ &= \mathrm{ggT}(5,3) = \mathrm{ggT}(3,2) = \mathrm{ggT}(2,1)\\ &= \mathrm{ggT}(1,0) = 1\end{aligned}$$

klassisch: 8 Schritte modern: 8 Schritte

34 und 21 sind zwei benachbarte Fibonacci-Zahlen.

Aufgabe 9

Implementiere die klassische Version des Algorithmus von Euklid in Python als Funktion mit dem Namen `ggt_alt(a, b)` und den formalen Parametern `a` und `b`. Die Funktion soll den ggT als Wert zurückgeben. Achte darauf, dass der Code syntaktisch korrekt ist. (pro Feld maximal ein Zeichen)

Aufgabe 9

```
def ggt_alt(a, b):
    a, b = abs(a), abs(b)
    while b > 0:
        if a < b:
            a, b = b, a
        a = a - b
    return a
```

Aufgabe 10

Implementiere die moderne Version des Algorithmus von Euklid in Python als Funktion mit dem Namen `ggt_neu(a, b)` und den formalen Parametern `a` und `b`. Die Funktion soll den berechneten ggT als Wert zurückgeben. Achte darauf, dass der Code syntaktisch korrekt ist. (pro Feld maximal ein Zeichen)

Aufgabe 10

```
def ggt_neu(a, b):
    a, b = abs(a), abs(b)
    while b > 0:
        a, b = b, a % b
    return a
```