

Aufgabe: Finde einen Algorithmus zur Berechnung des ggT .

Eingabe: Zwei natürliche Zahlen $i, j \in \mathbb{N}$

Ausgabe: Der größte gemeinsame Teiler von i, j : $ggT(i, j) = \max(a \mid i \bmod a = 0 \wedge j \bmod a = 0)$

1. Falls $i < j$, vertausche i und j . : Invariante $i \geq j$ wird hergestellt
2. $i := i - j$: Rechenschritt
3. Falls $i \neq 0$, mache weiter bei 1. : Abbruchbedingung
4. Ausgabe j .

Begründung der Korrektheit

Es liegt der Euklidische Algorithmus zugrunde. Das Verfahren beruht darauf, dass für alle $i > j \in \mathbb{N}$ gilt $ggT(i, j) = ggT(i - j, i)$. Durch Schritt 1 wird gewährleistet, dass die kleinere von der größeren Zahl abgezogen wird. Im Schritt 2 wird die größere beider Zahlen durch ihre Differenz ersetzt. In jedem Schleifendurchlauf verringert sich somit die Summe beider Zahlen. Daher muss nach einer endlichen Zahl von Schleifendurchläufen einmal der Fall $i = 0$ eintreten. Der Wert von j ist nach obiger Gleichung der ggT der beiden eingegebenen Zahlen.

Laufzeitanalyse

Die Laufzeit ist abhängig von der Eingabe. Bei jedem Schleifendurchlauf verringert sich die Summe beider Zahlen um mindestens 1. Bei Ausgabe des ggT beträgt die Summe beider Zahlen mindestens 1. Die Laufzeit lässt sich also abschätzen mit $T(i, j) \leq 2 \cdot (i + j - 1)$.

Beispiel: $i = 10, j = 1, T(i, j) = 2 \cdot (10 + 1 - 1) = 20$. Die Schleife wird insgesamt 10x durchlaufen, je Schleifendurchlauf gibt es zwei Vergleiche.

Speicherbedarf

Der Algorithmus benötigt zwei Speicherplätze für die Zahlen i und j , eventuell einen weiteren zur Durchführung des Tausches (1.) und der Subtraktion. Der Speicherplatzbedarf ist also unabhängig von der Eingabe; es werden 2 bzw. 3 Speicherplätze benötigt.

Anmerkungen

Wenn man einen Algorithmus beschreibt, so muss man zunächst definieren, was der Algorithmus tun soll. Dies wird durch die Beschreibung von Eingabe und Ausgabe eindeutig bestimmt. Den Algorithmus selbst formuliert man am besten in Schritten. Gewöhnlich nutzt man dafür sogenannten *Pseudocode*, das sind Rechen- und Vergleichsoperationen sowie Variablenzuweisungen in Verbindung mit Umgangssprache.

Soll man einen Algorithmus analysieren, so muss man seine *Korrektheit* nachweisen, und dass er terminiert. Korrektheit heißt, man muss zeigen, dass der Algorithmus das tut, was er tun soll, also eine korrekte Ausgabe liefert. *Termination* heißt, dass der Algorithmus in jedem Fall anhält und nicht etwa ewig läuft.

Ein weiterer Gegenstand der Analyse sind *Laufzeit* und *Speicherbedarf* des Algorithmus. Für die Laufzeit beschränkt man sich in der Regel auf die Anzahl der Vergleiche, in unserem Falle würde man zählen, wie oft die Schritte 1 und 3 durchlaufen werden. Üblicherweise betrachtet man den sogenannten *worst case*: den ungünstigsten Fall in dem im Verhältnis zur Eingabegröße die meisten Vergleiche benötigt werden. Für den Speicherbedarf zählt man für jeden Schritt die Anzahl der benötigten Speicherplätze und wählt das Maximum.