Der euklidische Algorithmus

Univ.-Prof. Dr. C. Fuchs

Unterlagen für Lehrende

30.11.2023

Grundsätzlich: Der ggT von a, b ist der größte gemeinsame Teiler von a und b. Der euklidische Algorithmus ist ein Verfahren zur Berechnung des ggTs.
Technologieeinsatz mit Sage:
Link zu SageCell: `https://sagecell.sagemath.org/`
Einige nützliche Sage-Befehle: `factor`, `gcd`, `xgcd`, `divisors`, `%`, `//`
Programm in Stichworten:
Definition, Wechselwegnahme, Division mit Rest (Existenz und Eindeutigkeit), Euklidischer Algorithmus, Korrektheit des Algorithmus (Analyse Teil 1), Laufzeitanalyse (Analyse Teil 2, u.a. Laufzeit hängt nur vom Quotienten ab, Quotienten sind immer minestens 1, Fibonacci-Zahlen mit Eigenschaften).

Seien a, b natürliche Zahlen. Wir bezeichnen mit T_a die Menge der Teiler von a bzw. mit T_b die Menge der Teiler von b, d.h. $T_a = \{t; t|a\}, T_b = \{t; t|b\}$. Dann ist $T_a \cap T_b$ die Menge der gemeinsamen Teiler. Sie ist wegen $1 \in T_a \cap T_b$ nicht leer und (da jeder Teiler von a höchstens a groß ist) durch $\min\{a, b\}$ beschränkt. Als Teilmenge von $\mathbb{N}$ hat $T_a \cap T_b$ daher ein größtes Element, das wir den größten gemeinsamen Teiler von a und b nennen (da es der größte gemeinsame Teiler von a und b ist) und mit $\text{ggT}(a, b)$ bezeichnen. Führe die Arbeitsaufträge 1 bis 5 aus.

(Bemerkung: In der Schule würde man zuerst die Primfaktorzerlegung von a und b berechnen. Damit kennt man dann sämtliche Teiler. Fasst man die Primfaktoren aufsteigend nach den vorhandenen Primzahlen zu Potenzen zusammen, so muss man nur zu jeder Basis die kleinere Primzahlpotenz aufmultiplizieren, um den ggT zu berechnen. Das Faktorisieren von Zahlen ist aber schwierig, wenn die Zahlen groß sind. Daher ist dieses Verfahren für die Praxis nicht geeignet. Zudem wird der ggT häufig verwendet, um den Hauptsatz der Zahlentheorie (von der eindeutigen Primfaktorzerlegung) zu beweisen. Beantworte Arbeitsauftrag 6.

Die Definition liefert einen naiven Algorithmus, um den ggT zu berechnen. Berechne nämlich alle Teiler von a und suche unter ihnen jene Teiler, die auch b teilen. Somit erhalten wir $T_a \cap T_b$. Suche schließlich in dieser Menge das größte Element. Dies läßt sich noch etwas verbessern: Teste beginnend mit $t = a$, ob t ein Teiler von a ist. Ist dies der Fall, so teste, ob t auch b teilt. Ist das der Fall, so ist $t = \text{ggT}(a, b)$. Andernfalls verkleinere t um 1 und beginne von vorne. Führe die Arbeitsaufträge 7 und 8 aus.

Diese Vorgangsweise ist für die Praxis ebenfalls ungeeignet. Vor über 2000 Jahren hatte Euklid eine geniale Idee. Diese Idee ist Grundlage für eine andere Vorgangsweise und ist vielleicht der 1. mathematische Algorithmus überhaupt. Dieser Algorithmus ist zudem extrem effizient und für moderne Anwendungen in unserer digitalen Welt von größter Bedeutung. Grundlage ist die folgende Aussage für $a, b \in \mathbb{N}$ mit $a \geq b$:

$$\text{ggT}(a, b) = \text{ggT}(a - b, b).$$

Bezeichenen wir mit d den ggT von a und b. Dann ist d ein gemeinsamer Teiler von a und b. Damit teilt d aber auch $a - b$, denn: $a = xd, b = yd$ und somit $a - b = xd - yd = (x - y)d$. Somit ist d ein gemeinsamer Teiler von $a - b$ und b und daher gilt $d \leq \text{ggT}(a - b, b)$. Bezeichnen wir mit d' den ggT

MAJA wird gefördert vom Bundesministerium für Frauen, Wissenschaft und Forschung:
Bundesministerium Frauen, Wissenschaft und Forschung

von $a-b$ und b. Mit einem ähnlichen Trick folgt, dass d' ein gemeinsamer Teiler von a und b ist. Daher gilt $d' \leq d$. Zusammengefasst gilt $d \leq d' \leq d$. Das zeigt die behauptete Gleichheit. Das von Euklid erdachte Verfahren ist die Methode der Wechselwegnahme. Bei der Berechnung von $\mathrm{ggT}(a,b)$ ziehen wir vom grösseren der beiden Zahlen die kleinere ab und machen dann mit der kleineren Zahl und der erhaltenen Differenz weiter. Bei diesem Verfahren wird stets eine der beiden Zahlen echt kleiner. Nach endlich vielen Schritten ist eine der beiden Zahlen 0 (das ist eine besondere Eigenschaft der natürlichen Zahlen, die wir uns als Leiter vorstellen können)! Der ggT von a und 0 ist klarerweise gleich a, d.h. $\mathrm{ggT}(a,0)=a$. (Hier verwenden wir, dass jede Zahl 0 teilt.) Führe die Arbeitsaufträge 9 und 10 aus.

Wie wir am Beispiel gesehen haben, wird in der Regel die kleinere Zahl mehr als einmal abgezogen. Fibonacci hatte die Idee, dies zu einem Schritt zusammenzufassen. Wir sehen uns die zugrundeliegende Aussage zunächst etwas genauer an: Sei $a,b \in \mathbb{N}$ mit $a \geq b$. Dann gibt es $q,r \in \mathbb{Z}$ mit $a=qb+r$ so, dass $0 \leq r < b$. Die Zahlen q,r mit diesen beiden Eigenschaften sind zudem eindeutig bestimmt. Wir sagen, dass r der Rest bei Division von a durch b ist und schreiben $r = a \bmod b$ dafür (in Sage ist $r = a\%b$). Hier die Begründung: Wir betrachten die Menge $\{a-bt; t \in \mathbb{Z}\}$. Diese Menge ist nicht leer und enthält somit eine kleinste natürliche Zahl r (das ist wieder eine charakteristische Eigenschaft von $\mathbb{N}$). Offenbar ist $0 \leq r < b$ und $r = a - qb$. Nehmen wir an, es gibt q', r' mit $a = q'b + r'$ und $0 \leq r' < a$. Dann gilt $a = qb + r = q'b + r'$ und somit $(q-q')b = r' - r$. Die linke Seite ist ein (echtes?) Vielfaches von b. Die rechte Seite liegt aber zwischen 0 und b, wobei b nicht in Frage kommt. Das geht aber nur, wenn die rechte Seite 0, d.h. $q = q'$ ist, und dann auch $r = r'$ gilt. Führe die Arbeitsaufträge 11 und 12 aus.

Wir setzen $r_{-1} = a$, $r_0 = b$ und berechnen sukkzessive $r_1 = r_{-1} - r_0 q_1$ (indem wir r_{-1} in der Form $r_{-1} = r_0 q_1 + r_1$ mit $0 \leq r_1 < r_0$ darstellen), $r_2 = r_0 - r_1 q_2$, etc. Im Schritt k berechnen wir $r_k = r_{k-2} - r_{k-1} q_k$. Wir hören auf, wenn 0 auftritt, d.h. wenn $r_{n+1} = r_{n-1} - r_n q_{n+1} = 0$ liefert. Der gesuchte ggT ist dann gleich r_n. (Für letzteres noch eimal zur Erinnerung: $\mathrm{ggT}(a,b) = \mathrm{ggT}(r_{-1}, r_0) = \mathrm{ggT}(r_0, r_1) = \mathrm{ggT}(r_1, r_2) = \cdots = \mathrm{ggT}(r_n, r_{n+1}) = \mathrm{ggT}(r_n, 0) = r_n$.) Führe die Arbeitsaufträge 13 und 14 aus.

Wir führen als nächstes nun die Laufzeitanalyse durch. Dabei helfen uns drei Überlegungen, die wir nacheinander formulieren und begründen werden.

Überlegung 1: Die Anzahl der Schritte des ggT hängt nur vom Verhältnis von a zu b, also vom Quotienten a/b, ab. Wir können also annehmen, dass der gesuchte ggT gleich 1 ist. Weiters sehen wir, dass das Problem nicht „skalierbar“ ist, d.h. wenn wir a und b mit einem Faktor e multiplizieren, ändert sich der ggT um denselben Faktor e, die Laufzeit des euklidischen Algorithmus bleibt aber gleich!
Wir begründen jetzt die Überlegung: Sei $a/b = c/d$. Dann gibt es eine ganze Zahl e mit $a = ce$ sowie $b = de$. Sei jetzt $r_1, r_2, \ldots$ die Folge, die der euklidische Algorithmus angewandt auf das Paar (a,b) ausgibt. Analog sei $r'_{-1} = c, r'_0 = d$ und $r'_1, r'_2, \ldots$ die Folge, die der euklidische Algorithmus angewandt auf das Paar (c,d) ausgibt. Da $r_{-1} = a = ce = r'_{-1}e$ und $r_0 = b = de = r'_0 e$, folgt $q_1 = q'_1$ sowie $r_1 = r_{-1} - r_0 q_1 = r'_{-1}e - r'_0 q_1 = (r'_{-1} - r'_0 q_1)e = r'_1 e$. Genauso sieht man $q_2 = q'_2, r_2 = r'_2 e, q_3 = q'_3, r_3 = r'_3 e, \ldots$.

Überlegung 2: Für $k = 1, \ldots, n$ gilt $g_n \geq 1$. Außerdem gilt $q_{n+1} \geq 2$. Wir können jetzt Extrembeispiele erzeugen. Wir geben uns dazu die Laufzeit n vor, setzen die q's kleinstmöglich und suchen nach a und b, sodass der euklidische Algorithmus in genau n Schritten den ggT bestimmt. Führe die Arbeitsaufträge 16 und 17 aus. Wir erhalten also allgemein $r_{n+1} = 0, r_n = 1, r_{n-1} = 2 \cdot r_n + r_{n+1} = 2, r_{n-2} = 1 \cdot r_{n-1} + r_n =$

MAJA wird gefördert vom Bundesministerium für Frauen, Wissenschaft und Forschung:

$2+1=3, r_{n-3}=1\cdot r_{n-2}+r_{n-1}=3+2=5,\ldots$. Die so erhaltene Folge $(1,2,3,5,\ldots)$ ist die Folge der Fibonacci-Zahlen $(F_2,F_3,F_4,F_5,\ldots)$. Somit ist $a=r_{-1}=F_{n+3}, b=r_0=F_{n+2}$. (Die Fibonacci-Zahlen treten also als worst-case bei der Laufzeitanalyse des euklidischen Algorithmus auf!)
Wir begründen jetzt die Überlegung: Die Folge der r_k ist strikt fallend und eine ganze Zahl, die nicht gleich 0 ist, ist ≥ 1. Daher ist die erste Aussage richtig. Angenommen es wäre $q_{n+1}=1$. Dann gilt $r_{n-1}=r_nq_{n+1}+r_{n+1}=r_n$, was nicht mit der Funktionsweise des euklidischen Algorithmus zusammenpasst. Somit ist $g_{n+1}\geq 2$.

Überlegung 3: Für $k=0,1,\ldots,n$ gilt $r_k\geq\theta^{n-k}$, wobei n so gewählt ist, dass $r_n=\text{ggT}(a,b)$ und $\theta=(1+\sqrt{5})/2$ den Goldenen Schnitt bezeichnet. Es folgt für $k=0$, dass $b=r_0\geq\theta^n$ ist. Somit $n\leq\log b/\log\theta$. Z.B. für $b=10^{100}$ ist $n\leq 479$. Wir sehen also, dass die Anzahl der Schritte nur logarithmisch in b ist, was einen extrem effizienten Algorithmus ergibt.
Mit Hilfe von Überlegung 1 können wir annehmen, dass $r_n=1$. Wir zeigen die Aussage durch vollständige Induktion nach n. Es gilt $r_n=1=\theta^0$. Zudem gilt $r_{n-1}=r_nq_{n+1}\geq 2>\theta$. Sei $n-2\geq k\geq 0$ und angenommen die Aussage ist mit Index $k+1$ und $k+2$ richtig. Dann gilt $r_k=r_{k+1}q_{k+2}+r_{k+2}\geq r_{k+1}+r_{k+2}\geq\theta^{n-k-1}+\theta^{n-k-2}=\theta^{n-k-1}(1+1/\theta)=\theta^{n-k}$. Somit ist die behauptete Ungleichung belegt.
Führe abschließend Arbeitsauftrag 18 aus.

Literatur:

1. A. Bartholomé, J. Rung, H. Kern: Zahlentheorie für Einsteiger – Eine Einführung für Schüler, Lehrer, Studierende und andere Interessierte. Vieweg+Teubner, 2010, https://doi.org/10.1007/978-3-8348-9650-6
2. C. Fuchs: Zahlentheorie, Vorlesungsskript, PLUS, 2022
3. C. Fuchs: Kryptologie, Vorlesungsskript, PLUS, 2022
4. G. Bard: Sage for Undergraduates, AMS, 2022

Der euklidische Algorithmus

Univ.-Prof. Dr. C. Fuchs

Handout

30.11.2023

Grundsätzlich: Der ggT von a, b ist der größte gemeinsame Teiler von a und b. Der euklidische Algorithmus ist ein Verfahren zur Berechnung des ggTs.
Technologieeinsatz mit Sage:
Link zu SageCell: `https://sagecell.sagemath.org/`
Einige nützliche Sage-Befehle: `factor`, `gcd`, `xgcd`, `divisors`, `%`, `//`
Programm in Stichworten:
Definition und Eigenschaften, Wechselwegnahme, Division mit Rest (Existenz und Eindeutigkeit), Euklidischer Algorithmus, Korrektheit des Algorithmus (Analyse Teil 1), Laufzeitanalyse (Analyse Teil 2, u.a. Laufzeit hängt nur vom Quotienten ab, Quotienten sind immer mindestens 1, Fibonacci-Zahlen mit Eigenschaften).

Arbeitsaufträge:

1. Welche Methoden zur Berechnung des ggTs gibt es?
2. Berechne den ggT von 123123 und 555555.
3. Berechne den ggT von 955354721 und 330435.
4. Berechne den ggT von 491891400 und 1138845708.
5. Berechne den ggT von 360^{360} und 540^{180} sowie von $2^{32} - 1$ und $3^8 - 2^8$.
6. Warum ist die Berechnung mit Hilfe der Primfaktorzerlegung in der Praxis nicht sinnvoll verwendbar?
7. Implementiere den naiven Algorithmus in Python.
8. Implementiere den verbesserten Algorithmus in Python.
9. Berechne die obigen ggTs durch „Wechselwegnahme“.
10. Implementiere die Methode der Wechselwegnahme in Python.
11. Dividiere 12121212 durch 11 mit Rest.
12. Dividiere 12345678 durch 12 mit Rest.
13. Bestimme mit dem euklidischen Algorithmus den ggT der folgenden Zahlenpaare $(1008, 840)$, $(481, 1755)$, $(2940, 1617)$.
14. Kürze jeweils und schreibe als gemischte Zahl:

$$\frac{3381821}{17759}, \quad \frac{48529591}{6186818}.$$

MAJA wird gefördert vom Bundesministerium für Frauen, Wissenschaft und Forschung:

15. Angenommen die Berechnung des ggT hat 20mal den Quotient 1 und als Ergebnis 1 ergeben. Wie lauten die ursprünglichen Zahlen?

16. Wie in der letzten Aufgabe aber mit 200 anstatt 20.

17. Sei $b = 12$. Ermittle a so, dass es $1, 2, 3, \ldots$ Divisionen benötigt, um den ggT zu bestimmen.

MAJA wird gefördert vom Bundesministerium für Frauen, Wissenschaft und Forschung:

Erkläre den euklidischen Algorithmus.

Was ist dir sonst zum euklidischen Algorithmus in Erinnerung geblieben?

Das hat mir gefallen. Das hat mir nicht gefallen.

MAJA wird gefördert vom Bundesministerium für Frauen, Wissenschaft und Forschung: