Bemerkung:

Die logarithmische Abhängigkeit der Schrittzahl von der Ausgangsgröße $a + b$ des Problems ist recht vorteilhaft und wird hier wie bei vielen algorithmischen Problemen, wie z.B. dem Sortieren, durch Aufspaltung in kleinere Aufgaben ähnlicher Art erreicht.

Dabei wird davon ausgegangen, daß der eigentliche Rechenaufwand pro Schritt (also die Auswertung von $b \bmod a$) konstant sei.

Allerdings ist diese implizite Annahme nicht ganz korrekt:

Wie wir später sehen werden, wächst dieser Aufwand (genau wie bei Addition und Multiplikation auch) mit $\lg(a + b)$. Der genaue Aufwand hängt von der Zahldarstellung und der entsprechenden Datenstruktur ab.

Herleitung des Erweiterten Euklidschen Algorithmus

Im Euklidischen Algorithmus wird jeweils aus den aktuellen Werten $a > 0$ und $b > a$ das Residuum $r = b - a * q < a$ berechnet. Wir bezeichnen nun die Ausgangswerte von a und b mit a_0 und b_0 und suchen jeweils Darstellungen

$$a = s_a * a_0 + t_a * b_0, \quad b = s_b * a_0 + t_b * b_0, \quad r = s_r * a_0 + t_r * b_0$$

Ganz am Anfang gelten diese Beziehungen mit $s_a = 1 = t_b$ und $s_b = 0 = t_a$. Aus $r = b - a * q$ folgt zudem, dass jeweils

$$s_r = s_b - q * s_a \quad \text{und} \quad t_r = t_b - q * t_a$$

Da die Koeffizienten bezüglich b_0 (nämlich t_a, t_b und t_r) uns letztlich nicht interessieren, ist die einzige Zusatzrechnung die Anweisung $s_r = s_b - q * s_a$ wobei allerdings das Ersetzen von (a, b) durch (r, b) mit durchgeführt werden muss. Bezeichnen wir jeweils s_a mit s und s_b mit v so ergibt sich die folgende Prozedur.

Lemma A.89 (Existenz und Berechnung von Inversen in $\mathbb{Z}_b$)

Die Zahl $a < b$ hat genau dann ein multiplikatives Inverses a^{-1} im Restklassenring $\mathbb{Z}_b$, wenn a und b **relativ prim** *sind, d.h. $GGT(a, b) = 1$. Dann gilt*

$$a^{-1} = s \bmod b \quad \textit{für} \quad 1 = s * a + t * b$$

Bemerkung

Bislang haben wir multiplikative Inverse von a unter den Potenzen $a^k \bmod b$ für $k = 0, 1, \ldots$ gesucht, was spätestens für $k = b - 2$ zum Erfolg führen muss (Satz A.54). Jetzt können wir den Euklidischen Algorithmus so erweitern, dass er den Koeffizienten s gleich mitberechnet und damit das Inverse a^{-1} von a mit einem Aufwand proportional zu $log_2 b$ ergibt.

Erweiterter Euklidischer Algorithmus:

Input: $a, b \in \mathbb{N}_+$ mit $0 < a < b$

$r := b \bmod a; \quad s = 1; \quad v = 0;$

`WHILE` $(0 \neq r)$

$q := (b - r)/a$

$b := a$

$a := r$

$t := v - q * s$

$v := s$

$s := t$

$r := b \bmod a$

Output: $a,\ s \bmod b$

Beispiel A.90 ($a = 16, b = 21$)

i	q	b	a	v	s	r
0	–	21	16	0	1	5
1	1	16	5	1	-1	1
2	3	5	1	-1	4	0

In Worten:

Der erweiterte Euklidische Algorithmus liefert uns $GGT(16, 21) = 1$ (der letzte Wert von a) und $s = 4$. Also existiert die Inverse von 16 in $\mathbb{Z}_{21}$ und ist gegeben durch 4. Die Probe ergibt tatsächlich

$$16 * 4 \bmod 21 = 64 \bmod 21 = 1\,.$$

Iterative Herleitung der Lösung

Offensichtlich ist $x = r$ eine Lösung der ersten Gleichung. Um deren Gültigkeit nicht zu verletzen dürfen wir ein beliebiges Vielfaches von m zu r addieren, also $x = r + q * m$. Dabei ist q so zu wählen, dass die zweite Gleichung erfüllt ist, d.h.

$$s = (r + q * m) \bmod n \;=\; [r \bmod n + (m \bmod n) * (q \bmod n)] \bmod n$$

und somit

$$(s - r) \bmod n \;=\; [(m \bmod n) * (q \bmod n)] \bmod n$$

Aus der Vorraussetzung, dass m und n relativ prim sind, ergibt sich nun die Existenz einer Inversen $c \in \mathbb{Z}$ von $m \bmod n$ so dass $c * m \bmod n = 1$. Multiplizieren wir die obige Gleichung mit diesem c, so erhalten wir mit Hilfe der Assoziativität in $\mathbb{Z}$ als mögliche Wahl für q

$$q = [c * (s - r)] \bmod n\,.$$

Daraus ergibt sich die folgende Aussage:

Bemerkung

Die Fähigkeit, modulare Inverse effizient zu berechnen, kann benutzt werden, um die nach dem Chinesischen Restsatz existierenden Lösungen von Kongruenzengleichungen zu finden.
Wir betrachten zunächst ein Paar von Gleichungen

$$x \bmod m = r \quad \textit{und} \quad x \bmod n = s\,,$$

wobei naturgemäß $r < m$ und $s < n$ sein müssen. Man sieht sofort, dass mit irgendeinem x auch alle ganzen Zahlen der Form $x + k * KGV(m, n)$ für beliebiges $k \in \mathbb{Z}$ Lösungen sind.
Nur falls $KGV(m, n) = m * n$ und äquivalenterweise $GGT(m, n) = 1$ kann man hoffen, dass es zwischen 0 und $n * m$ genau eine Lösung gibt.
Dies ist in der Tat der Fall, wie wir im folgenden herleiten werden.

Lemma A.91

Vorrausgesetzt $GGT(m, n) = 1$ *und* $c = (m \bmod n)^{-1}$, *dann ist die Zahl*

$$x = (r + [c * (s - r) \bmod n] * m) \bmod (m * n)$$

die einzige Lösung zwischen 0 *und* $n * m - 1$ *für die beiden Gleichungen*

$$x \bmod m = r \quad \textit{und} \quad x \bmod n = s\,.$$

Beispielrechnung

Für $m = 9, r = 3, n = 7$ und $s = 6$ erhalten wir die Gleichungen

$$x \bmod 9 = 3 \quad \text{und} \quad x \bmod 7 = 6\,.$$

Sie sind sicherlich lösbar, da $GGT(9,7) = 1$, ja 7 sogar eine Primzahl ist. Deswegen können wir die Inverse von $m \bmod n = 9 \bmod 7 = 2$ in $\mathbb{Z}_7$ einfacherweise nach dem kleinen Fermat'schen Satz A.54 auswerten.

$$2^{-1} = 2^{7-2} \mod 7 = 32 \bmod 7 = 4$$

Probe: $4 * 2 \bmod 7 = 1$.
Die Lösung ergibt sich nach der obigen Formel als

$$\begin{aligned} x &= (3 + [4 * (6-3) \bmod 7] * 9) \bmod 63 \\ &= (3 + 45) \bmod 63 \\ &= 48 \end{aligned}$$

Probe: $48 \bmod 9 = 3$ und $48 \bmod 7 = 6$ wie erwünscht.

Verallgemeinerung auf $n > 2$ Gleichungen

Betrachte ein System von Kongruenzen

$$x \bmod m_i = r_i < m_i \quad \text{für} \quad i = 1 \ldots n$$

unter der Vorraussetzung, dass die m_i paarweise relativ prim sind, d.h.

$$GGT(m_i, m_j) = 1 \quad \text{für} \quad 1 \le i < j \le n$$

Mit der Abkürzung $M = \prod_{j=1}^{n} m_j$ ergibt sich zunächst:

Lemma A.92

Für alle $i = 1 \ldots n$ ist das folgende Produkt relativ prim zu m_i

$$M_i = \prod_{j=0}^{i-1} m_j \prod_{j=i+1}^{n} m_j \;=\; M/m_i\,,$$

aber ein Vielfaches aller anderen m_j mit $j \neq i$. Es gilt also

$$GGT(M_i, m_j) \;=\; \begin{cases} 1 & \textit{falls} \quad j = i \\ m_j & \textit{falls} \quad j \neq i \end{cases}$$

und somit

$$M_i \bmod m_j \;=\; 0 \quad \textit{falls} \quad j \neq i\,.$$

Direkte Herleitung

Will man bei der Lösung jegliche Abhängigkeit von der Reihenfolge der Gleichungen vermeiden, kann man den folgenden direkten Ansatz benutzen:

$$x = (x_m * m + x_n * n) \bmod (n * m) \quad \text{mit} \quad x_m, x_n \in \mathbb{Z}$$

Daraus ergeben sich für x_m und x_n die Gleichungen

$$x \bmod m = (x_n * n) \bmod m = r \quad \text{und} \quad x \bmod n = (x_m * m) \bmod n = s$$

Mit $c_n < m$ die Inverse von n in $\mathbb{Z}_m$ and $c_m < n$ die Inverse von m in $\mathbb{Z}_n$ erhalten wir einfach

$$x_m = c_m * s < m * n \quad \text{und} \quad x_n = c_n * r < n * m$$

Hier erhält man für x zunächst einen Wert zwischen 0 und $2 * n * m$, von dem man gegebenenfalls einmal $n * m$ abziehen muss um im Interval $0, 1, \ldots n * m - 1$ zu landen.

Explizite Lösung

Nach obigem Lemma existieren Inverse $c_i < m_i$ von $(M_i \bmod m_i)$ in $\mathbb{Z}_{m_i}$, mit deren Hilfe wir eine Lösung direkt hinschreiben können:

$$\begin{aligned} x &= [r_1 * c_1 * M_1 + r_2 * c_2 * M_2 + \cdots + r_n * c_n * M_n] \bmod M \\ &= \left[\sum_{i=1}^{n} r_i * c_i * M_i\right] \bmod M \end{aligned}$$

$$\begin{aligned} \textit{Probe}: \quad x \bmod m_j &= \left[\sum_{i=1}^{n} r_i * c_i * M_i\right] \bmod M \bmod m_j \\ &= \left[\sum_{i=1}^{n} r_i * c_i * M_i\right] \bmod m_j \\ &= r_j * (c_j * M_i \bmod m_j) \\ &= r_j \end{aligned}$$