Ergänzungen zum Skript der Vorlesung

Diskrete Mathematik

Prof. Dr. Wolfgang Konen

Multiplikation großer Zahlen mit dem CRT

Man sucht zunächst teilerfremde Zahlen $m_1,\ldots,m_n$, deren Produkt eine Zahl größer als das größte erwartete Ergebnis liefert. Hier im Beispiel reichen 3 hohe zweistellige Zahlen, z.B. $97\cdot98\cdot99 \approx 1.000.000$, da das Produkt $220\cdot599$ im niedrigen 6-stelligen Bereich zu erwarten ist. Dann stellt man jede Zahl bezüglich ihrer Reste dar

	mod 97	mod 98	mod 99
220	26	24	22
599	17	11	5
220·599	26·17 = 54	24·11 = 68	22·5 = 11

Hierzu sind nur n Multiplikationen im zweistelligen Zahlbereich nötig **UND NICHT n^2** (!!) Gemäß dem Chinesischen Restsatz (CRT) sind alle Zahlen zwischen 0 und $97\cdot98\cdot99-1$ eindeutig durch ihre drei Reste bzgl. 97, 98 und 99 charakterisiert, also charakterisiert das Tripel

(54, 68, 11)

eindeutig die Produktzahl $220\cdot599$.

„Ja, aber“ werden Sie sagen, „man muss doch auch noch rücktransformieren!“ Und zum Rücktransformieren sind doch erhebliche Multiplikationen nötig. Nach Satz S1-7 (Explizite Lösung für CRT) hat man ja zu bilden

$$x = (a_1 \cdot M_1 \cdot N_1 + \ldots + a_n \cdot M_n \cdot N_n) \mod m$$

was in diesem Fall bedeutet:

$$\begin{aligned} x &= (54\cdot98\cdot99\cdot N_{97} + 68\cdot97\cdot99\cdot N_{98} + 11\cdot97\cdot98\cdot N_{99}) \mod m \\ &= (54\cdot Y_1 + 68\cdot Y_2 + 11\cdot Y_3) \mod m \end{aligned} \qquad (*)$$

mit bestimmten multiplikativen Inversen N_{97}, N_{98} und N_{99} und bestimmten (vermutlich großen) Zahlen Y_1, Y_2 und Y_3.

Warum die Rücktransformation keinen hohen Aufwand darstellt:

- Nicht nur die multiplikativen Inversen N_{97}, N_{98} und N_{99}, auch die Multiplikationen **(!!)** kann man vorwegberechnen:
  - Es gibt nur 97 Zahlen $a_1 \in \mathbf{Z}_{97}$ mit denen $Y_1 = 98 \cdot 99 \cdot N_{97}$ multipliziert wird.
  - Es gibt nur 98 Zahlen $a_2 \in \mathbf{Z}_{98}$, mit denen $Y_2 = 97 \cdot 99 \cdot N_{98}$ multipliziert wird.
  - Es gibt nur 99 Zahlen $a_3 \in \mathbf{Z}_{99}$, mit denen $Y_3 = 97 \cdot 98 \cdot N_{98}$ multipliziert wird.
  - Auch wenn Y_1, Y_2 und Y_3 große Zahlen sind, diese Multiplikationen kann man vorweg berechnen und in einer Tabelle ablegen. Es sind also nur 3 Table-Lookups in ≤99 großen Tabellen nötig. Auch die (mod m)-Berechnung kann man gleich mit durchführen und in der Tabelle ablegen >> Zahlen aus $\mathbf{Z}_m$.
  - Schließlich sind dann nur noch 3 Zahlen aus $\mathbf{Z}_m$ zu addieren und – falls Summe ≥ m – muss man einmal oder zweimal m abziehen.
  - Alle diese Operationen sind bestenfalls von **O(n)** und nicht $O(n^2)$.
- Manchmal braucht man auch gar nicht rücktransformieren, sondern kann mit dem Tripel (54, 68, 11) gleich weiterrechnen.

Zur Komplexität des Algorithmus Erweiterter Euklid

Wieviel Rekursionen, d.h. wieviel Zeilen in der Tabelle, können beim Algorithmus von Euklid maximal vorkommen? – Hierzu brauchen wir nur den Hinweg, also den „normalen“ Euklid (ohne x und y) betrachten, da die Zeilen beim Rückweg ja festliegen.

Behauptung: Der Algorithmus Euklid(n,m) braucht höchstens **2·ld(n)** rekursive Aufrufe. D.h. die Zahl der Tabellenzeilen wird nie größer als das Doppelte der Ziffernzahl in der Binärdarstellung der Zahl n.

Beispiel: Bei einer Zahl mit 100 Dezimalziffern sind das höchstens 2·100 ld(10) = 665 Zeilen.

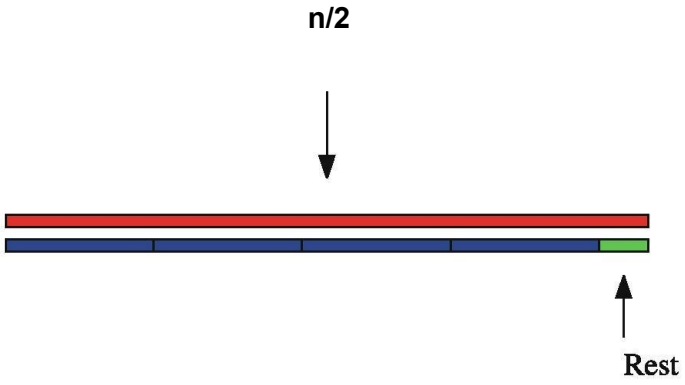

Abbildung 1: In die rote Zahl n passt die blaue Zahl m 4mal hinein. Der Rest ist kleiner als n/2.

Beweis: Aus Abbildung 1 macht man sich klar:

(1) $n \geq m + r$

Ferner gilt immer, denn der Rest ist aus $\mathbf{Z}_m$:

(2) $m > r$

Setzt man (2) in (1) ein, so folgt

(3) $n > r + r \quad \Leftrightarrow \quad r < n/2$

Dies ist die entscheidende Beobachtung, dass der Rest nie größer als n/2 wird.

Nach einer Runde wird m durch r, also eine Zahl < n/2 ersetzt. In der 2. Runde wandert dieses r auf den Platz für n und nun wird m durch das neue r, also eine Zahl < m/2 < n/2 ersetzt. Sicherlich sind jetzt beide Zahlen n und m kleiner als n/2.

Beispiel:

n		m		floor(n/m)
255		184		1
184	<255	71	<255/2	2
71	<255/2	42	<255/2	
…		…		

Nach 2·k Runden sind beide Zahlen kleiner als $n/2^k$. Setzen wir k = ld(n) ein, so folgt, dass beide Zahlen ≤ 1 sind, wir sind also sicher schon vorher fertig. ■

Literatur

- **[Vöcking08]** Vöcking, B.; Alt, H. et al. (Hrsg.) **Taschenbuch der Algorithmen,** Reihe: eXamen.press, 448 Seiten, 20€, Springer, Berlin Heidelberg, 2008. Ein Kapitel daraus von Friedrich Eisenbrand erklärt gut den Euklidschen Algorithmus und seine Komplexität. Alle Kapitel online verfügbar unter SpringerLink an FHK.