Primzahltests und Primzahlrekorde

Prof. Dr. Günter M. Ziegler
Institut für Mathematik
Technische Universität Berlin
Straße des 17. Juni Nr. 136
10623 Berlin

ziegler@math.tu-berlin.de

Die letzten Jahre haben uns eine Serie von neuen Primzahl-Rekorden beschert. So wurde im November 2005 von F. Bahr, M. Boehm, J. Franke, T. Kleinjung das RSA-640 Entschlüsselungsproblem gelöst: die Faktorisierung einer 193-stelligen Dezimalzahl. Im September 2006 wurde die bisher größte bekannte Primzahl gefunden, die Mersenne-Zahl

$$M = 2^{32.582.657} - 1,$$

mit insgesamt 9.808.358 Stellen. Möglicherweise stehen wir damit ganz kurz vor der Entdeckung einer Primzahl mit mehr als zehn Millionen Stellen, worauf ein Preisgeld von 100.000 US$ ausgesetzt ist.

Mersennesche Zahlen

Seit Januar 1996 läuft im Internet eine Suche nach immer größeren Mersenneschen Primzahlen. In dem verteilten Rechenprojekt unter dem Titel GIMPS („Great Internet Mersenne Prime Search“, www.mersenne.org), können Freiwillige übers Internet die GIMPS-Computerprogramme abrufen und „ihre“ Zahlen zum Testen zugeteilt bekommen, ihre PCs damit Sklavenarbeit leisten lassen, und die Rückmeldung übers Internet abliefern.

Marin Mersenne, 1588 – 1648 (Quelle: www-groups.dcs.st-and.ac.uk/~history/PictDisplay/Mersenne.html)

Zur Erinnerung: zu Ehren des französischen Mönches Marin Mersenne (1588 – 1648) heißen die Zahlen der Form $M_n = 2^n - 1$ *Mersennesche Primzahlen* — wenn sie prim sind. Dafür ist notwendig (schöne Übungsaufgabe aus der elementaren Zahlentheorie), dass n selbst prim ist. Aber hinreichend ist das nicht: $n = 11$ liefert das erste Gegenbeispiel. Im Jahr 1644 behauptete Mersenne, dass M_n für $n = 2, 3, 5, 7, 13, 17, 19, 31, 67, 127$ und 257 prim sei, aber keine andere Primzahl unter 257 (womit er exakt fünfmal danebengelegen hat).

Mersennesche Primzahlen sind ziemlich selten: Man weiß nicht, ob es unendlich viele Mersennesche Primzahlen gibt, man kennt inzwischen die ersten 39 und nur fünf weitere, darunter die neu gefundene $M_{32.582.657}$, die derzeit auch die größte bekannte Primzahl ist.

Dass man Zahlen mit fast zehn Millionen Stellen effektiv auf Primalität testen kann, ist die eigentliche wissenschaftliche (und programmiererische) Höchstleistung hinter dem neuen Rekord — dass $n = 32.582.657$ prim sein muss, ist ja nur eine klitzekleine Aufwärmübung für den neuen Rekord.

Primalitätstests

Nun weiß man seit Kurzem, dass es exakte Primzahltests gibt, die in Polynomzeit laufen — siehe [1]. Diese stellen einen theoretischen Durchbruch dar, sind aber für den Einsatz in der Praxis (noch) nicht geeignet. Im GIMPS-Projekt wird für jedes prime n eine Kaskade von klassischeren Tests durchlaufen, die unter www.mersenne.org/math.htm sehr schön und kapierbar beschrieben

werden. Zur algorithmischen Primzahltheorie empfehlen mir Experten das Buch [2]. Aus Computeralgebra-Perspektive finden sich Primzahltests (und sehr viel mehr Spannendes) in [3]. In *Phase I* sucht man nach kleinen Primteilern q von $2^n - 1$. Diese müssen (wieder eine hübsche Übungsaufgabe) $q \equiv 1 \bmod 2n$ und $q \equiv \pm 1 \bmod 8$ erfüllen. Mithilfe eines auf solche Faktoren zugeschnittenen „Sieb des Eratosthenes“ werden dann Primteiler von M_n bis ca. 40.000 erkannt. Dabei kann ausgenutzt werden, dass Teilbarkeitstests für Zahlen vom Typ $2^n - 1$ in Binärarithmetik sehr effektiv durchgeführt werden können.

In *Phase II* wird dann ein Spezialfall der sogenannten $(p-1)$-Methode von Pollard (1974) verwendet, mit der man Faktoren $q = 2kn + 1$ finden kann, für die $q - 1 = 2kn$ aus vielen kleinen Primfaktoren besteht, oder aber (in einer verbesserten Version) bis auf einen etwas größeren Primfaktor stark zusammengesetzt ist: Wenn man q sucht, so dass alle Primfaktoren kleiner als B sind, so bildet man dafür das Produkt $E := \prod_{p<B} p$ aller Primzahlen, die kleiner als B sind, und berechnet dann $x := 3^{E2n}$. Im ggT von $x - 1$ und $2^n - 1$ fängt man dann den gesuchten Teiler von $2^n - 1$.

www.mersenne.org

Erst in *Phase III* verwendet man dann ein Verfahren, mit dem man sicher entscheiden kann, ob $2^n - 1$ prim ist, den sogenannten Lucas-Lehmer-Test (1878, 1930/1935) für Mersenne-Zahlen: M_n ist genau dann prim, wenn $\ell_{n-1} \equiv 0 \bmod M_n$ gilt, wobei die ℓ_k durch $\ell_1 = 4$ und $\ell_n = \ell_{n-1}^2 - 2$ rekursiv definiert werden. Um das effektiv zu berechnen, muss man riesige Zahlen schnell modulo $2^n - 1$ quadrieren. Dazu werden die Zahlen in große Blöcke unterteilt, und dann arbeitet man mit Spezialversionen einer schnellen Fouriertransformation („Fast Fourier Transform“, FFT), in diesem Fall mit einer FFT bezüglich einer irrationalen Basis, die von Richard Crandell und Barry Fagin (*Mathematics of Computation* 1994) eingeführt wurde. Auf einer der WWW-Seiten des *Mathematica*-Projekts mathworld.wolfram.com, die die aktuelle Rekordmeldung verbreiten, wird suggeriert, GIMPS würde mit einer *Mathematica*-Implementierung arbeiten, aber das ist eine arge Dehnung der Tatsachen. (Es hat nur Crandall die Methode auch für die Primzahltests von *Mathematica* implementiert.) In der Tat arbeitet GIMPS mit hochoptimiertem Assembler-Code, aus Prozessorarchitekturgründen in Gleitkommaarithmetik, deren Fehler getrennt erkannt und aufgefangen werden müssen.

Primalität und Faktorisierung

Phasen I und II des GIMPS-Verfahrens spucken also im Fall von zusammengesetztem M_n wirklich Teiler aus — *wenn* sie welche finden —, die dritte und entscheidende Phase aber nicht mehr. Die Antwort heißt da dann nur noch „zusammengesetzt!“, ohne einen expliziten (Prim-)Teiler als Beweis. Es wird also ein Primalitätstest durchgeführt, aber kein vollständiges Faktorisierungsverfahren.

Und das ist auch gut so: Nicht einmal für den Spezialfall von Mersenne-Zahlen kennt man effektive Verfahren zum Faktorisieren. Ein Verfahren, mit dem man *beliebige* Zahlen mit ein paar Hundert Stellen faktorisieren könnte, wäre interessant und bedrohlich, weil die kryptographischen Verfahren, die die Sicherheit von Online-Banking und Internet garantieren sollen, darauf beruhen, dass das Faktorisieren und verwandte Probleme (wie die Berechnung von „diskreten Logarithmen“) offenbar schwer sind.

RSA

Ein Beispiel dafür ist das von Ron Rivest, Adi Shamir und Leonard Adleman 1978 publizierte Verschlüsselungsverfahren „mit öffentlichen Schlüsseln“, das sich inzwischen in fast jedem elementaren Zahlentheorie-Lehrbuch findet, gleichzeitig aber auch in der Praxis vielfältig zum Einsatz kommt — siehe die Homepage www.rsa.com der Firma von Rivest, Shamir und Adleman.

Die Sicherheit des Verfahrens gegen unerlaubtes Entschlüsseln hängt davon ab, dass es mit heutiger Technologie sehr schwer ist, Produkte von Zahlen mit 150 – 200 Stellen in ihre Primfaktoren zu zerlegen. Die Firma „RSA Securities“ hatte sogar Preise von insgesamt über 630.000 US$ auf Beispielprobleme (de.wikipedia.org/wiki/RSA_Factoring_Challenge) ausgesetzt.

Für das Faktorisieren der Zahl „RSA-576“ hat Jens Franke von der Universität Bonn im Dezember 2003 ein Preisgeld von 10.000 US$ kassiert. Im November 2005 konnte er in Teamarbeit die Faktorisierung von RSA-640 vermelden. Dabei ging es um die Zahl

3107418240490043721350750035888567930037346022842727545720161948823206
4405180815045563468296717232867824379162728380334154710731085019195485
2900733772482278352574238645401469173660247765234609

mit 193 Dezimalziffern bzw. 640 Binärziffern (bits). Dafür gab es 10.000 US$. Diese Zahl hat die Faktoren

1634733645809253848443133883865090859841783670033093231218111085238933310010450815121211816751157 9

und

$$1900871281664822113126851573935413975471896789968515493666638539088027103802104498957191261465571$$

(mit je 97 Ziffern), und die sind prim — was wiederum mit den aktuellen Methoden ganz leicht zu zeigen ist. Franke verwendete dabei das „General Number Field Sieve (GNFS)“. Dieses wurde von Lenstra, Lenstra, Manasse & Pollard 1990 eingeführt, und hat eine Laufzeit von $\exp\left(O\left(\sqrt[3]{n \log n}\right)\right)$ für n-stellige Zahlen; es ist also nicht ganz polynomial, aber *fast*. Unter Verwendung des GNFS wurden auch schon die kleineren Testprobleme von RSA-100 bis RSA-512 geknackt.

Im Mai 2006 hat RSA Secturity den „RSA Factoring Challenge“ als beendet erklärt. Die Schwierigkeit des Problems sei inzwischen hinreichend geklärt — und dies, obwohl es ja bisher keinen *Beweis* gibt, dass das Faktorisieren von Zahlen praktisch oder theoretisch wirklich schwer ist.

Die weiteren Preise des „RSA Factoring Challenge“ werden also nicht mehr ausgezahlt: die reichten ursprünglich bis zu 20.000 US$, für die Faktorisierung des Testproblems RSA-2048.

Das hat allerdings Jens Franke et al. nicht aufgehalten: Mit Hilfe von Supercomputern in Bonn, an der EPFL Lausanne und am NTT in Japan gelang ihnen die vollständige Faktorisierung von $M_{1039} = 2^{1039} - 1$, also einer Zahl mit 1039 bits (die allerdings kein RSA-Testproblem war).

Und es gibt noch mehr aktuelle Rekorde, die sich ebenfalls aufs Faktorisieren beziehen: Unter anderem versucht man eben Mersenne-Zahlen nicht nur auf Primalität zu untersuchen, sondern auch vollständig in Primfaktoren zu zerlegen. So will man im „Cunningham project“ (homes.cerias.purdue.edu/~ssw/cun/) die Zahlen der Form $b^n \pm 1$ für $b = 2, 3, 5, 6, 7, 10, 11$ und 12 bis zu hohen Werten von n vollständig faktorisieren. Als Spezialfälle enthält dies die Mersenne-Zahlen $M_n = 2^n - 1$, die nur für primes n selbst Primzahlen sein können, und die Fermat-Zahlen $F_n = 2^{2^n} + 1$. (Übungsaufgabe: Eine Zahl $2^m + 1$ kann nur dann prim sein, wenn m eine Zweierpotenz ist.) Soweit wir wissen, ist F_n nur für $n = 0, 1, 2, 3$ und 4 prim. Euler selbst hat gezeigt, dass $F_5 = 4294967297$ durch 641 teilbar ist. Das verteilte Internet-Projekt NFSNET (www.nfsnet.org) ist dabei extrem erfolgreich, und liefert fast jeden Monat einen neuen Eintrag in die Ergebnisliste. Die Erfolge basieren dabei auf dem „Special Number Field Sieve (SNFS)“ — einer schnelleren Spezialversion des GNFS, die nur für spezielle Zahlen, eben etwa vom Typ $b^n \pm 1$, anwendbar ist.

Rekordjagd

Die Rekordjagd geht weiter. Die „Electronic Frontier Foundation“ (www.eff.org/) hat schon im Jahr 2000 einmal 50.000 US$ für die erste Primzahl mit einer Million Stellen ausgezahlt. Für die Identifikation einer Primzahl mit mehr als 10 Millionen Dezimalstellen hat sie 100.000 US$ ausgesetzt. Dies heizt die Stimmung an, und das GIMPS-Projekt sucht Mitstreiter, die ihre Computer für die Rekordjagd einsetzen wollen.

Viele arme kleine PCs werden also mit Zahlen gefüttert und mit Primzahltests und mit Zerlegungsverfahren gequält werden, nur damit Herrchen vielleicht einen Teil des Ruhms (und des Preisgeldes) einkassieren kann.

Literaturverzeichnis

[1] F. Bornemann, *Ein Durchbruch für „Jedermann“*, in Computeralgebra-Rundbrief Nr. 32, 2003, S. 8 – 14.

[2] R. Crandall und C. Pomerance, *Prime Numbers — A Computational Perspective*, Springer-Verlag, New York, 2001.

[3] J. von zur Gathen und J. Gerhard, *Modern Computer Algebra*, Cambridge University Press, 2. Auflage, Cambridge, 2003.