Mit dem Computer auf der Jagd nach Zahlen mit großer Teilersumme

Sebastian Flothow* Jan Gampe† Steffen Reith‡ Jörn Steuding§

7. Mai 2012

Perfekte Zahlen

Eine natürliche Zahl kann sehr viele, aber auch sehr wenige Teiler besitzen. Beispiele für den letzteren Fall sind die Primzahlen wie etwa $2, 3, 5, 7, 11, 13, \ldots$, welche außer eins und sich selbst keine Teiler besitzen. Primzahlen sind die multiplikativen Bausteine der natürlichen Zahlen und trotz ihrer einfachen Definition existieren viele ungelöste Fragestellungen zu eben diesen, so etwa die Riemannsche Vermutung bzgl. ihres Auftretens oder auch die Goldbachsche Vermutung über die Darstellbarkeit gerader Zahlen als Summe von zwei Primzahlen. Andere natürliche Zahlen $n > 1$ heißen zusammengesetzt und auch diese bergen interessante Geheimnisse. Bereits bei den alten Griechen findet sich der Begriff der *perfekten* Zahl (oder auch *vollkommenen* Zahl) für solche natürlichen Zahlen n, deren Teiler sich zum Wert $2n$ aufsummieren. Die erste perfekte Zahl ist somit

$$6 = \tfrac{1}{2}(1 + 2 + 3 + 6)$$

und die nächsten weiteren sind $28, 496, 8128$. Euklid beobachtete, dass diese Beispiele perfekter Zahlen alle eine gemeinsame Struktur besitzen und bewies entsprechend, dass Zahlen der Form

$$2^{p-1}(2^p - 1)$$

stets perfekt sind, wenn $2^p - 1$ eine Primzahl ist. Euler zeigte, dass alle *geraden* perfekten Zahlen von dieser Form sind (Beweis im Appendix). Primzahlen der Form $2^p - 1$ werden nach dem mittelalterlichen Mönch und Mathematiker Marin Mersenne benannt, wobei es nicht schwierig zu zeigen ist, dass dabei der Exponent p selbst wieder eine Primzahl sein muss. Zahlen

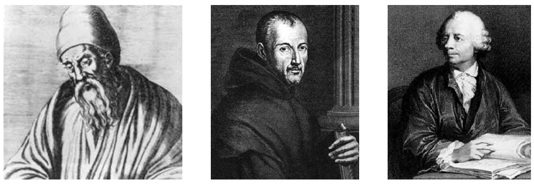

Abbildung 1: von links nach rechts: Euklid (ca. 325-265), Marin Mersenne (1588-1648), Leonhard Euler (1707-1783)

dieser speziellen Form lassen sich schnell mit dem Lucas–Lehmer–Test auf Primalität prüfen: Hierzu bilde man iterativ

$$s_1 = 4, \quad s_{k+1} = s_k^2 - 2$$

und teste, ob s_{p-1} durch $2^p - 1$ teilbar ist:

$$4 \mapsto 2 \cdot \mathbf{7} \mapsto 2 \cdot 97 \mapsto 2 \cdot \mathbf{31} \cdot 607 \mapsto \ldots$$

*Hochschule RheinMain, Fachbereich DCSM, Unter den Eichen 5, 65195 Wiesbaden, sebastian@flothow.de

†Hochschule RheinMain, Fachbereich DCSM, Unter den Eichen 5, 65195 Wiesbaden, mail@jangampe.de

‡Hochschule RheinMain, Fachbereich DCSM, Unter den Eichen 5, 65195 Wiesbaden, Steffen.Reith@hs-rm.de

§Universität Würzburg, Institut für Mathematik, Am Hubland, Campus-Nord, Emil-Fischer-Str. 40, 97074 Würzburg, steuding@mathematik.uni-wuerzburg.de

Natürlich bietet es sich an, jeweils nur den Rest von s_{k+1} bei Division durch $2^p - 1$ zu bilden. Aufgrund dieses schnellen Primzahltests bilden die Mersenneschen Primzahlen seit Jahrzehnten die Rekordhalter für die größten bekannten Primzahlen. Heute sind 47 perfekte Zahlen bekannt; die größte ergibt sich aus der im GIMPS-Projekt gefundenen aktuell größten bekannten Mersenneschen Primzahl $2^{43\,112\,609} - 1$ nach Euklids Muster als

$$2^{43\,112\,608}(2^{43\,112\,609} - 1);$$

diese Zahl besitzt 25 956 378 Dezimalstellen[1]. Interessanterweise ist bis heute keine einzige *ungerade* perfekte Zahl bekannt. Allerdings weiß man, dass eine solche mindestens fünfhundert Dezimalstellen besitzen und aus mindestens acht verschiedenen Primteilern aufgebaut sein müsste.

Die Riemannsche Vermutung

Die Riemannsche Vermutung ist eines der sieben Millenniumsprobleme der Mathematik. Dieses ungelöste Problem handelt ursprünglich von den Nullstellen der so genannten Riemannschen Zetafunktion und lässt sich in sehr unterschiedliche Behauptungen gleichbedeutend umformulieren. Beispielsweise ist die Riemannsche Vermutung dazu äquivalent, dass die Anzahl $\pi(x)$ der Primzahlen $p \leq x$ gemäß

$$\pi(x) = \int_2^x \frac{\mathrm{d}u}{\log u} + \mathrm{Fehler}(x)$$

mit einem Fehlerterm der Ordnung $x^{\frac{1}{2}+\epsilon}$ wächst, wobei ϵ eine beliebige positive Größe ist und log den natürlichen Logarithmus bezeichnet. Eine solche Asymptotik mit einem schlechteren Fehlerterm wurde zuerst vom erst fünfzehnjährigen Universalgenie Carl Friedrich Gauß im Jahre 1792 vermutet und ca. einhundert Jahre später durch Hadamard und (unabhängig) de La Vallée-Poussin bewiesen. Ein verbesserter Fehlerterm, wie der aus der Riemannschen Vermutung resultierende, hätte wichtige Anwendungen in den verschiedensten Disziplinen, in denen Primzahlen auftreten, wie etwa in der Zahlentheorie und der Kryptographie.

Eine weitere interessante Umformulierung der Riemannschen Vermutung gelang Guy Robin 1984. Es bezeichne $\sigma(n)$ die Summe der Teiler von n. Dann zeigte Robin [5], dass die Riemannsche Vermutung genau dann wahr ist, wenn die Ungleichung

$$\frac{\sigma(n)}{n} < e^\gamma \log(\log n) \qquad \text{für} \quad n \geq 5041$$

mit $\gamma = \lim_{N\to\infty}(\sum_{n\leq N} \frac{1}{n} - \log N) = 0,577\ldots$ der Euler-Mascheroni-Konstante (bzw. $e^\gamma = \exp(\gamma) = 1,781\ldots$) besteht. Tatsächlich gilt ohne Annahme der Riemannschen Vermutung die nur unwesentlich schwächere Ungleichung

$$\frac{\sigma(n)}{n} < e^\gamma \log(\log n) + \frac{\frac{7}{3} - e^\gamma \log(\log 12))}{\log(\log n)}$$

für alle natürlichen Zahlen $n \neq 1, 2, 12$. Jeff Lagarias [4] entfernte noch die Konstante e^γ und bewies, dass die Riemannsche Vermutung äquivalent zu der Ungleichung

$$\sigma(n) \leq h(n) + \exp(h(n))\log(h(n))$$

für beliebige natürliche n ist, wobei $h(n) = 1 + \frac{1}{2} + \ldots + \frac{1}{n}$ die n-te harmonische Zahl bezeichnet.

Nun wird die Riemannsche Vermutung wohl kaum mit der Robinschen Ungleichung oder auch der von Lagarias bewiesen werden; höchstens lässt sie sich für eine Menge natürlicher Zahlen numerisch verifizieren. Und skeptischere Charaktere könnten hoffen, mit Hilfe dieser Ungleichungen die Riemannsche Vermutung zu widerlegen. Wie ist dabei zu verfahren, wenn die Riemannsche Vermutung auf diese Art für viele natürliche Zahlen geprüft bzw. ein Gegenbeispiel gesucht werden soll?

[1] siehe www.mersenne.org

Brutale Gewalt

Viele Vermutungen in der Zahlentheorie haben die Form: *Jede natürliche Zahl n besitzt die Eigenschaft $E(n)$,* womit E ein so genanntes (Boolesches) Prädikat ist. Ein solches Prädikat liefert „wahr“, wenn die getestet Zahl n die Eigenschaft $E(n)$ hat und sonst „falsch“.

Lässt sich die Frage, ob eine Zahl n die Eigenschaft E hat, mit Hilfe eines (effizienten) Algorithmus formulieren, dann ergibt sich ein einfaches Rechenverfahren, um alle natürlichen Zahlen bis zu einer gegebenen Grenze m auf die Eigenschaft E zu überprüfen:

```
for n = 1 to m do
 if (E(n) = "falsch") then
   gebe "E gilt im Intervall nicht" aus;
   terminate;
 endif
endfor;
gebe "E gilt im Intervall" aus.
```

Dieser primitive Ansatz liefert respektable Ergebnisse sofern sich $E(n)$ schnell berechnen lässt, und wird als „brute-force“-Methode bezeichnet. Mit diesem Vorgehen kann eine solche Vermutung sicherlich nicht für alle (unendlich vielen) natürlichen Zahlen komplett bewiesen werden, wohl aber lässt sich die Aussage für eine gewisse (endliche) Teilmenge der natürlichen Zahlen überprüfen; eventuell kann so eine Vermutungen durch ein Gegenbeispiel widerlegt werden. Dieser Ansatz ist insbesondere durch die aktuell wachsende Verbreitung von parallelen Rechnersystemen interessant. So besitzen die üblichen PC-Prozessoren schon heute bis zu 24 unabhängige Rechnenkerne, und die Entwicklung von Graphikkarten erlaubt es, hunderte von Berechnungen gleichzeitig durchführen zu können. Der Innovationsdruck in der Computerbranche bietet hier also die Möglichkeit, mit verhältnismäßig günstiger „Hardware von der Stange“ umfangreiche Berechnungen für die Zahlentheorie auszuführen.

Im Fall der Robinschen Ungleichung ist dieser Ansatz besonders einfach umsetzbar, in dem die natürlichen Zahlen in kleine Intervalle zerlegt werden, um diese dann parallel auf die Gültigkeit der Ungleichung zu testen. Jedes der Teilintervalle kann dabei auf einem völlig unabhängigen Rechensystem geprüft werden. Dieser Ansatz wurde an der Hochschule RheinMain im Rahmen einer Lehrveranstaltung des Masterstudiengangs Informatik umgesetzt.

Superabundante Zahlen

Zur Verringerung des Rechenaufwandes beim Brute force-Ansatz ist es sinnvoll den Bereich der potentiellen Kandidaten für ein Gegenbeispiel möglichst stark einzugrenzen. So ist etwa für natürliche Zahlen n mit wenigen Teilern (etwa Primzahlen) die Robinsche Ungleichung offensichtlich erfüllt, gleiches gilt für perfekte Zahlen n, welche ja $\sigma(n) = 2n$ erfüllen. Im Falle von

$$\frac{\sigma(n)}{n} > 2$$

nennen wir n eine *abundante* Zahl, wobei *abundant* soviel wie *reichhaltig* bedeutet. Es ist nämlich $\frac{\sigma(n)}{n}$ groß, falls n viele Teiler besitzt. Tatsächlich dürfen wir hier das Wort „Teiler“ ohne weiteres durch „Primteiler“ ersetzen. Die Funktion $n \mapsto \frac{\sigma(n)}{n}$ ist nämlich multiplikativ; es genügt also, sie auf den Primzahlpotenzen auszurechnen und deren Produkte zu bilden:

$$\frac{\sigma(n)}{n} = \prod_{p|n} (1 + p^{-1} + \ldots + p^{-\nu_p(n)})$$

mit der Primfaktorzerlegung $n = \prod_{p|n} p^{\nu_p(n)}$. Darüberhinaus heißt n *superabundant*, wenn

$$\frac{\sigma(m)}{m} < \frac{\sigma(n)}{n} \qquad \text{für} \quad m < n$$

gilt. Die ersten superabundanten Zahlen sind somit

$$1, 2, 4, 6, 12, 24, 36, 48, 60, 120, 180, \ldots \;;$$

dies ist die Folge A004394 in Neil Sloanes *The On-Line Encyclopedia of Integer Sequences* (OEIS) [6]. Amir Akbary und Zachary Friggstad [1] zeigten kürzlich, dass das kleinste Gegenbeispiel zu Robins Ungleichung, wenn überhaupt existent, eine superabundante Zahl sein muss, was den notwendigen Rechenaufwand beim Brute force-Ansatz extrem verringert.

Der Computer

Obwohl die Robinsche Ungleichung (oder auch die Version von Lagarias) einfach erscheint und zur empirischen Überprüfung der Riemannschen Vermutung einlädt, ist die Verifikation für große n im Allgemeinen sehr aufwendig. Tatsächlich benötigt die Berechnung der Teilersumme $\sigma(n)$ für ein gegebenes n dessen Primfaktorzerlegung. Diese ist jedoch, nach heutigem Kenntnisstand, bei großen natürlichen Zahlen recht aufwändig zu berechnen – eine Tatsache, auf der viele moderne kryptographische Verfahren beruhen.

Wie löst man also das Problem, große Zahlen faktorisieren zu müssen, um ihre Teilersumme zu bestimmen? Keith Briggs [2] vermeidet diese rechenintensive Hürde, indem er die zu untersuchenden superabundanten Zahlen gleich in Form ihrer Primfaktorzerlegung findet. Ausgehend von ebendieser in der Gestalt

$$n = \prod_p p^{e_p} = 2^{e_2} \cdot 3^{e_3} \cdot 5^{e_5} \cdot \ldots \cdot q^{e_q}$$

gelten für superabundante Zahlen nämlich die folgenden, von Erdös entdeckten, Eigenschaften superabundanter Zahlen:

i) Für die Exponenten gelten $e_2 \geq e_3 \geq e_5 \geq \ldots \geq e_q$. Insbesondere tritt in einer superabundanten Zahl jede Primzahl bis zu einer gewissen Schranke mindestens einmal auf.

ii) Wenn $1 < j < i \leq q$, dann gilt $|e_i \lfloor e_j \log_i j \rfloor| \leq 1$, wobei $\log_i$ den Logarithmus zur Basis i bezeichne.

iii) Für alle $i \geq 2$ gilt: $i^{e_i} < 2^{e_2+2}$.

iv) Der letzte Exponent e_q ist stets gleich eins, außer bei den superabundanten Zahlen 4 und 36.

Diese Eigenschaften stellen notwendige, aber nicht hinreichende Bedingungen für superabundante Zahlen dar. Sie genügen jedoch, um viele andere Zahlen auszuschließen und so eine überschaubare Menge von Kandidaten zu bilden. Briggs [2] bildet aus dieser Beobachtung folgenden Algorithmus:

1. Beginne mit dem Term: 2^{e_2} mit $e_2 = 1$.

2. Erweitere den Term immer weiter um Primfaktoren, solange die oben genannten Eigenschaften erfüllt sind und füge diese Terme der Liste von Kandidaten hinzu.

3. Erhöhe e_2 um eins und beginne von vorn, bis zu einer beliebigen Obergrenze für e_2.

4. Bringe die Kandidatenliste in numerische Reihenfolge.

5. Filtere nun aus der Liste von Kandidaten die superabundanten Zahlen durch Berechnung der Teilersumme.

Dieses Verfahren lässt sich technisch gut umsetzen. Es zeigt sich jedoch schnell, dass obwohl etliche Zahlen durch die Eigenschaften i)–iv) ausgesiebt werden, immer noch recht viele nicht-superabundante Kandidaten erzeugt werden, und einem handelsüblichen Computer geht schnell der Speicher ob dieser vielen unliebsamen Kandidaten aus. Deshalb ist es vorteilhafter, auf existierende Listen superabundanter Zahlen zurück zu greifen; die schon genannte OEIS [6] hilft mit dem Wissen um die ersten Million superabundanter Zahlen aus.

Größenordnungen

Bei dieser Strategie zur Überprüfung der Robinschen Ungleichung und der Riemannschen Vermutung offenbart sich folgendes ernstes Problem. Ein moderner handelsüblicher PC ist in der Lage, mit Operanden bis zu einer Größe von $2^{64} \approx 10^{19}$ zu rechnen. Dieses Niveau erreicht bereits die 116-te superabundante Zahl. Die nachstehende Tabelle vergleicht den Index einer der Größe nach geordneten Liste superabundanter Zahlen s_n mit ihrem Wert:

Index n	s_n
10	120
50	1163962800
100	$\approx 10^{16.5}$
1.000	$\approx 10^{149.9}$
10.000	$\approx 10^{1378.4}$
100.000	$\approx 10^{12146}$
1.000.000	$\approx 10^{103082}$

Oberhalb dieser technisch bedingten Grenze müssen die Operanden in für den Prozessor noch handhabbare Einheiten zerlegt werden. Auch die Umsetzung der arithmetischen Operationen muss überdacht werden: Je größer die Zahlen werden, desto aufwändiger gestaltet sich beispielsweise ihre Multiplikation. Die üblichen aus der Schule bekannten Verfahren werden bei solch großen Zahlen zu langsam und andere Algorithmen wie die Karatsuba-Multiplikation oder Verfahren auf der Basis der „*Fast-Fourier-Transformation*" sind zu bevorzugen.

Um mit solch großen Zahlen schnell arbeiten zu können, verwenden wir Zahlenbibliotheken wie GMP (*GNU Multi-Precision Library*) oder die MPIR (*Multiple Precision Integers and Rationals*). Diese Zahlenbibliotheken können mit Zahlen von bis zu 41 Milliarden Dezimalstellen arbeiten, zukünftige Versionen sogar mit einer Quadrillion Dezimalstellen, wogegen unser verwendeter Prüfbereich mit 103082 Dezimalstellen verhältnismäßig winzig ist.

Hase und Igel

Robins Ungleichung kann als Wettlauf zweier Funktionen, nämlich $\frac{\sigma(n)}{n}$ und $e^\gamma \log(\log n)$, verstanden werden. Zur vereinfachten Darstellung stellen wir die Formel um:

$$\frac{\frac{\sigma(n)}{n}}{e^\gamma \log(\log n)} \leq 1 \qquad \text{für} \quad n \geq 5041.$$

Der Quotient auf der linken Seite nähert sich für die ersten 1.000.000 superabundanten Zahlen immer weiter der Eins an – der Zählerterm gewinnt also zunehmend an Boden. Abbildung 2 zeigt dieses Verhalten mit einer logarithmisch skalierten horizontalen Achse. Das Wachstum

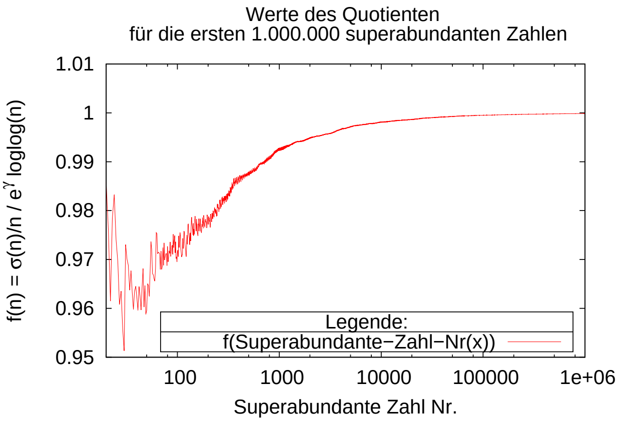

Abbildung 2: Robins Aussage: Nach der 20. superabundanten Zahl wird der Quotient stets unterhalb der Eins liegen!

des Zählerterms nimmt rapide ab, und es stellt sich die Frage, ob das Wachstum dennoch ausreichen wird, um irgendwann, bei einer sehr großen Zahl, doch noch den Nenner zu übertrumpfen. Man wünscht sich, stichprobenartig die 10^7-te oder die 10^8-te superabundante Zahl ansehen können, um mehr zu erfahren. Mit den bekannten Methoden lässt sich diese jedoch leider nur über ihre superabundanten Vorgänger in Erfahrung bringen.

Abbildung 2 zeigt, dass besagter Quotient sich nicht monoton verhält. Es ist zu beobachten, dass zu Beginn im Durchschnitt unserer Daten nur etwa jede zwanzigste superabundante Zahl einen neuen Rekord in der Annäherung zur Eins aufstellt. Der Abstand zu jedem neuen Rekordhalter steigt allerdings mit der Größe der superabundanten Zahlen - der Graph wird glatter.

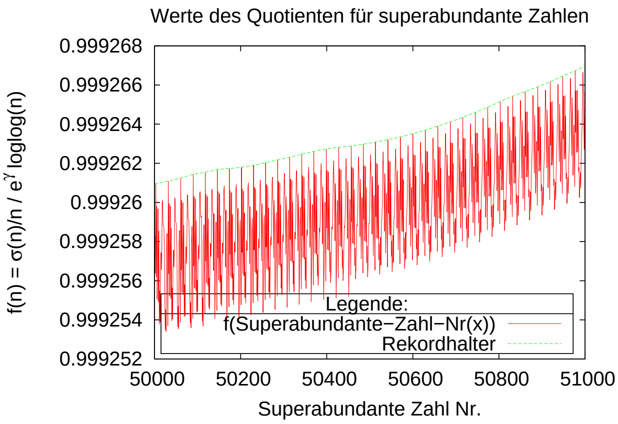

Abbildung 3: Die Funktion aus Abbildung 2 oszilliert stark.

Am Ende bleiben von 1.000.000 superabundanten Zahlen nur 25.000 Rekordhalter, die es also wert sind genauer untersucht zu werden. Diese Zahlen sind Beispiele so genannter extrem abundanter Zahlen genannt. Eine Zahl n heißt dabei *extrem abundant*, wenn

$$\frac{\sigma(n)}{n \log \log n} > \frac{\sigma(m)}{m \log \log m}$$

für $10.080 < m < n$. Gegenüber den superabundanten Zahlen unterscheiden sich diese extrem abundanten Zahlen wesentlich weniger in ihrer jeweiligen Primfaktorzerlegung. Mit dieser Beobachtung besteht die Hoffnung, einen neuen Algorithmus zu designen, der beliebig große extrem abundante Zahlen erzeugt, was zugleich ein Fortschritt hinsichtlich der Überprüfung der Robinschen Ungleichung mit sich brächte.

Grenzen des Machbaren

Je größer die zu untersuchenden Zahlen sind, desto mehr Zeit benötigen die damit verbundenen arithmetischen Rechenoperationen. Eine Programmanalyse, das sogenannte Profiling, gibt Aufschluss darüber, welche Operationen die meiste Rechenzeit benötigen. Dabei beobachtet wir bei unseren Berechnungen, dass knapp zwei Drittel der Zeit für Multiplikationen aufgebracht werden und ein Drittel das Notieren des Ergebnisses kostet. Würde man sich dabei auf das Notwendigste beschränken (Ausgabe von „Ja“ oder „Nein“), wäre auch dieser Aspekt zu vernachlässigen. Wie bereits erwähnt, basiert der einzige bekannte Algorithmus zur Bestimmung superabundanter Zahlen auf der Kenntnis aller kleineren superabundanten Zahlen. Noch aufwändiger ist die nachfolgende Aufgabe, nämlich die Teilersumme einer Zahl mit über 60 Millionen Dezimalstellen (wie etwa für die milliardste superabundante Zahl) zu bestimmen, wobei die Primfaktorzerlegung glücklicherweise bekannt ist. Für eine Rechenoperation benötigt man pro Dezimalstelle ca. acht Bytes Speicher für Zwischenrechnungen und das Ergebnis. Ein hochklassiger Heim-PC kommt in diesen Größenordnungen von mehreren hundert Megabyte Arbeitsspeicherbedarf pro Zwischenrechnung bereits an seine natürlichen Grenzen. Die Prüfung der billionsten superabundanten Zahl wird aber auch schon für Großrechner zu einer kaum zu bewerkstelligen Aufgabe: Der Arbeitsspeicher reicht nicht mehr aus, um eine einzige Variable zu halten; also muss alles schrittweise von Sekundärspeichern (z.B. Festplatten) nachgeladen werden. Und passen die Zahlen nicht mehr in den Hauptspeicher, führt dies zu einem Vielfachen an Laufzeit. Hier besteht also Bedarf, einen neuen effizienteren Algorithmus zu finden, um diese gigantischen Zahlen untersuchen zu können. Oder man vertraut auf Geduld und das *Mooresche Gesetz*, nachdem sich die Leistungsfähigkeit von Prozessoren und die Kapazität des Hauptspeichers alle ca. 24 Monate verdoppelt.

Appendix: Beweis des Satzes von Euklid und Euler

Jede gerade perfekte Zahl n ist von der Gestalt $n = 2^{p-1}(2^p - 1)$ mit Primzahlen p und $2^p - 1$. Um dies zu beweisen, verifiziert man zuerst für Zahlen n besagter Gestalt die Gleichung $\sigma(n) = 2n$. Ferner gilt für $n = 2^k m$ mit einem ungeraden m auf Grund der Multiplikativität von $\frac{\sigma(n)}{n}$

$$\sigma(n) = \sigma(2^k)\sigma(m) = (2^{k+1} - 1)\sigma(m).$$

Nehmen wir nun noch an, dass n perfekt ist, so gilt ferner

$$\sigma(n) = 2n = 2^{k+1}m.$$

Aus diesen beiden Identitäten ergibt sich

$$\frac{m}{\sigma(m)} = \frac{2^{k+1} - 1}{2^{k+1}}.$$

Da rechts ein gekürzter Bruch steht, gilt somit $m = (2^{k+1} - 1)\ell$ und $\sigma(m) = 2^{k+1}\ell$ für eine natürliche Zahl ℓ. Wäre nun $\ell > 1$, so besäße m sicherlich die Teiler $m, \ell, 1$, und es folgte

$$\sigma(m) \geq m+\ell+1 = 2^{k+1}\ell+\ell+1 > 2^{k+1}\ell = \sigma(m),$$

ein Widerspruch. Also ist $\ell = 1$ und $n = 2^k(2^{k+1} - 1)$ mit $\sigma(2^{k+1} - 1) = 2^{k+1}$. Letzteres impliziert, dass $2^{k+1} - 1$ eine Primzahl der Exponent $p = k + 1$ für sich bereits prim sein muss.

Literatur

[1] A. Akbary, Z. Friggstad, Superabundant numbers and the Riemann hypothesis, *American Mathematical Monthly* **116** (2009), 273-275

[2] K. Briggs, Abundant numbers and the Riemann hypothesis, *Experimental Mathematics* **15** (2006), 251-256

[3] R. Crandall und C. Pomerance, Prime Numbers – A Computational Perspective, 2nd edition, Springer, 2005

[4] J. C. Lagarias, An elementary problem equivalent to the Riemann hypothesis, *American Mathematical Monthly* **109** (2002), 534-543

[5] G. Robin, Grandes valeurs de la fonction somme des diviseurs et hypothèse de Riemann, *J. Math. Pures Appl.* **63** (1984), 187-213

[6] N. J. A. Sloane, The On-Line Encyclopedia of Integer Sequences, http://oeis.org/