

# Das Sieb des Eratosthenes

(Eratosthenes von Kyrene ca. 276-194 v.Chr.)

Jochen Ziegenbalg, Karlsruhe

Jede natürliche Zahl  $n$  besitzt die „trivialen“ Teiler 1 und  $n$  selbst; jede von 1 verschiedene natürliche Zahl besitzt also mindestens zwei Teiler. Zahlen, die genau diese beiden trivialen Teiler besitzen, nennt man *Primzahlen*. Nach dieser Definition wird die Zahl 1 also nicht zu den Primzahlen gerechnet. Dies hat gute Gründe; einer davon ist, dass sonst der Fundamentalsatz der Zahlentheorie nicht gelten würde.

Die Primzahlen sind einer der ältesten und interessantesten Untersuchungsgegenstände der Mathematik. Sie stellen u.a. die Bausteine dar, aus denen die natürlichen Zahlen aufgebaut sind. Der **Fundamentalsatz der Zahlentheorie** besagt:

Jede natürliche Zahl  $n$  ( $n > 1$ ) ist als Produkt von Primzahlen darstellbar:  
 $n = p_1 \cdot p_2 \cdot \dots \cdot p_s$ . Abgesehen von der Reihenfolge der Faktoren ist diese Darstellung eindeutig.

Auch für andere Zahlensysteme oder algebraische Systeme sind Primzahlen, Primelemente oder dem Primzahlbegriff nachgebildete Begriffe von zentraler Bedeutung. Eine faszinierende Eigenschaft der Primzahlen ist die Unregelmäßigkeit, mit der sie in der Zahlenreihe auftreten. Gesetzmäßigkeiten in der Primzahlenreihe zu entdecken, war schon immer eine wichtige Forschungsrichtung in der Mathematik.

Schon im Altertum war man bestrebt, einen möglichst guten Überblick über die Primzahlen zu gewinnen. Euklid zeigte, dass es unendlich viele Primzahlen gibt. Der griechische Mathematiker *Eratosthenes von Kyrene* gab das folgende Verfahren an, um alle Primzahlen bis zu einer bestimmten vorgegebenen Zahl  $n$  zu bestimmen. Es sei hier am Beispiel  $n = 20$  erläutert.

1. Schreibe alle Zahlen von 1 bis 20 auf:

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

2. Streiche die Zahl 1 (sie wird aus guten Gründen nicht zu den Primzahlen gerechnet):

~~1~~ 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

3. Unterstreiche die Zahl 2:

~~1~~ 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

4. Streiche alle echten Vielfachen von 2; also die Zahlen 4, 6, 8, 10, 12, 14, 16, 18 und 20:

~~1~~ 2 3 ~~4~~ 5 ~~6~~ 7 ~~8~~ 9 ~~10~~ 11 ~~12~~ 13 ~~14~~ 15 ~~16~~ 17 ~~18~~ 19 ~~20~~

5. Unterstreiche die erste freie (d.h. noch nicht unterstrichene oder gestrichene) Zahl; in diesem Fall also die Zahl 3:

~~1~~ 2 3 ~~4~~ 5 ~~6~~ 7 ~~8~~ 9 ~~10~~ 11 ~~12~~ 13 ~~14~~ 15 ~~16~~ 17 ~~18~~ 19 ~~20~~

6. Streiche aus den verbleibenden Zahlen alle echten Vielfachen von 3; also die Zahlen 9 und 15:

~~1~~ 2 3 ~~4~~ 5 ~~6~~ 7 ~~8~~ ~~9~~ ~~10~~ 11 ~~12~~ 13 ~~14~~ ~~15~~ ~~16~~ 17 ~~18~~ 19 ~~20~~

7. Unterstreiche die kleinste freie Zahl; in diesem Fall also die Zahl 5:

~~1~~ 2 3 ~~4~~ 5 ~~6~~ 7 ~~8~~ ~~9~~ ~~10~~ 11 ~~12~~ 13 ~~14~~ ~~15~~ ~~16~~ 17 ~~18~~ 19 ~~20~~

8. Streiche aus den verbleibenden Zahlen alle echten Vielfachen der Zahl 5; da die in Frage kommenden Zahlen 10, 15 und 20 bereits gestrichen sind, tritt in diesem Fall (Maximum = 20) keine Veränderung auf.
9. Setze das Verfahren sinngemäß so lange fort, bis jede der Zahlen entweder unterstrichen oder gestrichen ist.

~~1~~ 2 3 ~~4~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ ~~10~~ 11 ~~12~~ 13 ~~14~~ ~~15~~ ~~16~~ 17 ~~18~~ 19 ~~20~~

10. Ende des Verfahrens. Die unterstrichenen Zahlen sind die Primzahlen zwischen 1 und 20.

Durch dieses Verfahren werden, wenn man so will, also genau die Primzahlen „ausgesiebt“. Man nennt das Verfahren deshalb auch das *Sieb des Eratosthenes* bzw. kurz das *Siebverfahren*.

#### Aufgaben:

- (a) Führen Sie das Siebverfahren von Hand für die natürlichen Zahlen von 1 bis 200 durch.
- (b) Geben Sie eine allgemeine Beschreibung des Siebverfahrens, die von der Zahl 20 unabhängig ist; die Obergrenze sei allgemein mit  $a$  bezeichnet.
- (c) Zeigen Sie: Ist die Zahl  $a$  zerlegbar, z.B.  $a = x \cdot y$  (mit von 1 verschiedenen Faktoren  $x$  und  $y$ ), so ist einer der Faktoren kleiner oder gleich  $\sqrt{a}$ .
- (d) Aufgabenteil (c) hat zur Folge, dass das Siebverfahren *erheblich* verkürzt werden kann, denn man ist mit dem Streichen der Vielfachen schon fertig, wenn die unterstrichene Zahl  $\sqrt{a}$  erreicht hat. Formulieren Sie den Algorithmus so, dass diese Verbesserung der „Laufzeiteffizienz“ realisiert wird.

#### Bemerkungen:

(1.) Gelegentlich kann man lesen, dass das Sieb des Eratosthenes dazu dient, *die* Primzahlen (d.h. *alle* Primzahlen) zu ermitteln. Ein Blick auf den Algorithmus genügt aber, um festzustellen, dass er nur dann funktionieren kann, wenn man sich von vornherein auf einen *endlichen* Zahlenabschnitt beschränkt (im Beispiel: die natürlichen Zahlen von 1 bis 20). Das Sieb des Eratosthenes liefert also stets nur die Primzahlen bis zu einer bestimmten, von vorn herein festzulegenden, oberen Grenze. Diese Grenze lässt sich jedoch durch mehrere „Läufe“ des Verfahrens immer weiter nach oben verschieben. Die (unendliche) Menge der Primzahlen wird durch das Sieb des Eratosthenes also als „potentiell“ unendliche Menge erschlossen. Es sei an dieser Stelle an die außerordentlich weitsichtige Formulierung von Euklid erinnert: *Es gibt mehr Primzahlen als jede vorgelegte Anzahl von Primzahlen* (vgl. [Ziegenbalg, 2002, S. 51]).

(2.) Obwohl die eingangs geschilderte Version durchaus noch einige Beschleunigungsmöglichkeiten zulässt, ist das Siebverfahren ein sehr langsamer Algorithmus. Gerade aufgrund seiner Langsamkeit wurde er lange Zeit benutzt, um die Geschwindigkeit von Computern zu messen, denn schnelle Algorithmen laufen u.U. so schnell, dass man eine Zeitmessung nicht sinnvoll vornehmen kann. Die renommierte amerikanische Computerzeitschrift BYTE benutzte jahrelang ein auf dem Sieb des Eratosthenes basierendes Verfahren in diesem Sinne für sogenannte „benchmark tests“ (Geschwindigkeitstests für Hard- und Software). Das Sieb des Eratosthenes ist also im zweifachen Sinne klassisch zu nennen: im ursprünglichen Sinne von Eratosthenes zur Ermittlung von Primzahlen und neuerdings auch noch als benchmark test. Das von BYTE verwendete Programm ist ohne weiteres in verschiedene Programmiersprachen zu übertragen; es war jedoch im Hinblick auf Ergebnis und Ausgabemeldung fehlerhaft. Das Programm war somit zu überhaupt nichts nützlich, außer als Messinstrument für die Laufzeitgeschwindigkeit verschiedener Computersysteme.

#### Eine interaktive Simulation zum Sieb des Eratosthenes:

<http://www.ziegenbalg.ph-karlsruhe.de/materialien-homepage-jzbg/Sieb-des-Eratosthenes/Sieb-des-Eratosthenes-Simulation.htm>

## Programme zum Sieb des Eratosthenes:

In den folgenden beiden Programmen wird i.w. das (absichtlich) nicht auf Effizienz getrimmte Programm aus der Zeitschrift *BYTE* in den Computeralgebra Systemen *Mathematica* und *Maxima* (letzteres aus dem *Open-Source-Bereich*) nachgebildet (vgl. obige Bemerkungen) – allerdings ohne die dortigen sachlichen Fehler zu übernehmen.

Die Anweisung „Schreibe alle Zahlen ... auf“ wird umgesetzt, indem man sie als Elemente in eine Liste einfügt (im Sinne des Datentyps „List“ in der Programmiersprache LISP und einigen Computeralgebra Systemen). Die Anweisung „eine Zahl streichen ...“ wird dadurch realisiert, dass man den entsprechenden Listeneintrag auf Null setzt.

Implementierung im Computeralgebra System *Mathematica*:

```
SieveOfEratosthenes[UpperLimit_] :=  
  Module[{L = Table[t, {t, 1, UpperLimit}], i = 2, k},  
    L = ReplacePart[L, 0, 1];  
    While[i*i <= UpperLimit,  
      k=i+i;  
      While[k <= UpperLimit,  
        L = ReplacePart[L, 0, k];  
        k = k+i];  
      i = i+1];  
    Return[Select[L, Positive ] ] ]
```

*Comment from the Help-Browser of Mathematica:*

`ReplacePart[expr, new, n]` yields an expression in which the *n*-part of *expr* is replaced by *new*.

Ein konkreter Aufruf des obigen Programms (mit dem Parameter `UpperLimit = 80`) ergibt:

```
SieveOfEratosthenes[80]  
{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79}
```

Implementierung im Computeralgebra System *Maxima*:

```
Eratosthenes(UpperLimit) :=  
  block([E, i, k],  
    E : makelist(j, j, 1, UpperLimit),  
    E[1] : 0,  
    i : 2,  
    while i*i <= UpperLimit do  
      (k : i+i,  
        while k <= UpperLimit do  
          (E[k] : 0,  
            k : k+i),  
          i : i+1),  
    E : delete(0, E),  
    E);
```

Ein Aufrufbeispiel:

```
Eratosthenes(50);  
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]
```

## Gesamtdarstellungen in Buchform

Ziegenbalg J. / Ziegenbalg O. / Ziegenbalg B.: Algorithmen – von Hammurapi bis Gödel; Harri Deutsch Verlag, Frankfurt am Main 2007, Seite 71 ff

Ziegenbalg J.: Elementare Zahlentheorie - Beispiele, Geschichte, Algorithmen; Verlag Harri Deutsch, Frankfurt am Main 2002

Erste Hinweise zu *Maxima*:

<http://www.ziegenbalg.ph-karlsruhe.de/materialien-homepage-jzbg/Manuskripte/Erste-Hinweise-zu-Maxima.pdf>

## Email Adresse und Homepage des Autors

[ziegenbalg@ph-karlsruhe.de](mailto:ziegenbalg@ph-karlsruhe.de)

<http://www.ziegenbalg.ph-karlsruhe.de>