

Primfaktorzerlegung und Primzahltests

Maximilian Scholz
Technische Universität Hamburg-Harburg

9. Juli 2014

Zusammenfassung

Mit dieser Ausarbeitung im Rahmen des Proseminars Mathematik will ich eine Einleitung in die Welt der Primzahlen im Allgemeinen und der Primfaktorzerlegung im besonderen schaffen.

Ich beginne mit einer Einleitung zu den Primzahlen und zeige anhand von Beispielen, wie sich die Problematik der Primfaktorzerlegung auf verschiedene Wege angehen lässt. Hierzu beleuchte ich nicht nur die Algorithmen im einzelnen sondern zeige auch, welche Werkzeuge benutzt werden und erkläre auch diese detaillierter, da mir diese Herleitungen in der Fachliteratur oft zu kurz kommen.

Letztendlich ist mein Ziel Verständnis in einem sehr begrenzten Bereich zu schaffen anstatt in die Breite zu gehen, denn nur durch wirkliches Verständnis können neue Entdeckungen wie der zum Beispiel der AKS-Algorithmus gemacht werden.

Inhaltsverzeichnis

1	Einführung in Primzahlen	3
2	Sieb des Eratosthenes	3
2.1	Idee	3
2.2	Beispiel	4
2.3	Mathematik	5
2.3.1	Funktionsweise	5
2.3.2	Beweis	6
3	Pollard Rho Methode	6
3.1	Geburtstagsproblem	6
3.2	Hase Igel Algorithmus	7
3.2.1	Beispiel	7
3.2.2	Mathematik	9
3.3	Pollard Rho Algorithmus	9
3.3.1	Kongruenz modulo p	9
3.3.2	Idee	9
3.3.3	Beispiel	9
3.3.4	Mathematik	9
3.4	Komplexität	9
4	Fazit	10

1 Einführung in Primzahlen

Die wichtigsten Dinge die es zu Primzahlen im Bezug auf diesen Text gibt lassen sich wie folgt zusammenfassen:

- Definition: Jede natürliche Zahl > 1 , die nur von sich selber und 1 geteilt wird, ist eine Primzahl. Dies lässt sich auch so beschreiben, dass eine Primzahl nur beim Teilen durch 1 und sich selber keinen Rest hat.
- Der griechische Mathematiker Euklid hat bewiesen, dass es unendlich viele Primzahlen gibt. Würde dies nicht gelten, könnte es zu Problemen bei modernen Kryptographischen Verfahren kommen, da diese auf immer größer werdenden Primzahlen beruhen.
- Ebenfalls von Euklid kommt der Beweis dafür, dass sich jede natürliche Zahl als Produkt von Primzahlen darstellen lässt, wobei man hierfür die triviale Multiplikation mit der 1 hinzufügen muss, die keine Primzahl ist. Später fand Gauss heraus, dass diese sogenannte Zerlegung bis auf die Reihenfolge der einzelnen Elemente eindeutig ist.
Dies ist besonders interessant weil es uns ermöglicht zusammengesetzte Zahlen (also Zahlen, die nicht prim sind) daran zu erkennen, dass wir einen Faktor ungleich der Zahl oder 1 gefunden haben. Außerdem birgt dies die Grundlage zu manchen Angriffen auf Kryptografische Verfahren wie RSA, auf die in diesem Text allerdings nicht weiter eingegangen werden.

Dieser Text wird sich im weiteren Verlauf vor allem mit der Zerlegung von Zahlen in deren Primfaktoren befassen.

2 Sieb des Eratosthenes

Das Sieb des Eratosthenes ist ein etwa 2000 Jahre altes Verfahren, um alle Primzahlen in einem gegebenen Zahlenbereich zu finden.

2.1 Idee

Das Sieb des Eratosthenes bestimmt alle Primzahlen N , indem es alle zusammengesetzten Zahlen streicht, daher der Name: Sieb.

Das Abbruchkriterium ist ein schönes Beispiel wie unnötiger Aufwand vermieden werden kann.

2.2 Beispiel

Bevor die genaue Funktionsweise behandelt wird, hier einmal ein Beispiel für die Funktionsweise der Methode.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25				

Tabelle 1: Eratosthenes 1

Zu Beginn werden alle Zahlen im Bereich, in dem nach Primzahlen gesucht werden soll aufgeschrieben.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25				

Tabelle 2: Eratosthenes 2

Die Zahlen 0 und 1 sind per Definition keine Primzahlen, wir können sie somit streichen.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25				

Tabelle 3: Eratosthenes 3

Die erste nicht gestrichene Zahl ist 2. Gleichzeitig ist 2 die erste Primzahl. Da alle Zahlen die durch 2 teilbar sind nicht prim sind, können wir alle geraden Zahlen streichen.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25				

Tabelle 4: Eratosthenes 4

Die nächste nicht gestrichene Zahl ist 3. Somit ist auch 3 eine Primzahl. Wieder werden alle Vielfachen von 3 gestrichen, da sie durch 3 teilbar, und somit nicht prim, sind.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25				

Tabelle 5: Eratosthenes 5

Die 4 ist schon gestrichen, da sie durch 2 teilbar ist. Die nächste nicht gestrichene Zahl ist 5. Damit ist 5 die nächste Primzahl und alle Vielfachen von 5 werden gestrichen.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25				

Tabelle 6: Eratosthenes 6

Da nur die Zahlen bis 25 betrachtet werden, ist die Methode an dieser Stelle fertig. Mit $5 = \sqrt{25}$ haben wir das Abbruchkriterium erreicht und damit alle möglichen Teiler der Zahlen < 25 abgearbeitet. Die übrigen Zahlen, die grün unterlegt sind, sind Primzahlen.

2.3 Mathematik

2.3.1 Funktionsweise

Die Methode des quadratischen Siebes lässt sich wie folgt beschreiben:

1. Wähle eine natürliche Zahl $n > 1$. Dies ist die obere Grenze der Zahlen, in denen nach Primzahlen gesucht wird. In dem Beispiel ist $n = 25$.
2. Die kleinste noch nicht gestrichene Zahl m mit $2 \leq m$ wird die aktuelle Zahl. Das sind im Beispiel 2, 3 und 5.
3. Solange $m^2 \leq n$ gilt, streiche alle Vielfachen $c \cdot m$ ($c \in \mathbb{N}$) mit $m^2 \leq cm \leq n$. Im Beispiel durch die rote Färbung zu erkennen.
4. Übrig bleiben alle Primzahlen zwischen 0 und n . Im Beispiel durch die grüne Färbung zu erkennen.

Interessant ist zum einen das Abbruchkriterium, also wieso reicht es $m \leq \sqrt{n}$ zu betrachten um alle Primzahlen bis n zu finden. Außerdem sollte die intuitive Funktionsweise des Streichens von Vielfachen nicht als Beweis gesehen werden, dass die Methode des Siebens wirklich in jedem Fall funktioniert. Für diese beiden Teile werden im folgenden Beweise gezeigt.

2.3.2 Beweis

Beide Teile lassen sich zusammen beweisen:

Der kleinste Teiler einer zusammengesetzten Zahl n ist eine Primzahl p . Es gilt $p > 1$.

Da $p|n$ gilt, gilt auch $\frac{n}{p}|n$. Das Ergebnis von $\frac{n}{p}$ ist natürlich auch ein Teiler von n , da es mit p multipliziert n ergibt.

Da p der kleinste Teiler ist, gilt $p \leq \frac{n}{p}$, da jeder andere Teiler mindestens so groß wie p sein muss. Daraus folgt:

$$p^2 \leq n$$

Alle zusammengesetzten Zahlen $n < N$ werden also beim Sieben mit einer Zahl p , für die gilt $p \leq \sqrt{n}$, gestrichen.

Die übrigen Zahlen müssen also Primzahlen sein.

3 Pollard Rho Methode

3.1 Geburtstagsproblem

Die Idee des Geburtstagsproblems ist die Frage, wie hoch die Wahrscheinlichkeit auf einem Geburtstag mit N Personen ist, dass zwei am gleichen Tag Geburtstag haben.

Eine Möglichkeit das Problem einfacher zu machen ist, stattdessen das inverse Problem zu betrachten, also wie hoch die Wahrscheinlichkeit (P) ist, dass kein Geburtstag doppelt vorkommt.

Bei einem Besucher ist die Wahrscheinlichkeit

$$P(1) = \frac{356}{356}$$

da alle 356 Tage zur Wahl stehen. Bei zwei Besuchern ist die Wahrscheinlichkeit

$$P(2) = \frac{356}{356} \cdot \frac{364}{365}$$

bei drei Besuchern gilt

$$P(3) = \frac{356}{356} \cdot \frac{364}{365} \cdot \frac{363}{365}$$

Im Allgemeinen gilt:

$$P(N) = \frac{365 \cdot 364 \dots (365 - N + 1)}{365^N}$$

Da wir große Primzahlen betrachten werden interessiert uns auch das Verhalten des Geburtstagsproblems für große N . Beim Betrachten von vielen Geburtstagen mit N Personen, liegt, für große N , die Zahl der Wiederholungen die man durchschnittlich braucht um einen doppelten Geburtstag zu erhalten bei:

$$\sqrt{\frac{\pi \cdot N}{2}}$$

3.2 Hase Igel Algorithmus

Der Hase Igel Algorithmus wird benutzt, um Zyklen in Folgen zu finden. Die Idee ist, die Folge mit zwei Zeigern zu durchlaufen, dem Hase und dem Igel, die auf dem gleichen Feld starten. Falls es einen Zyklus in der Folge gibt, werden sich Hase und Igel irgendwann wieder treffen, da der schnellere Hase den langsameren Igel einholt und überrundet.

3.2.1 Beispiel

In diesem Beispiel ist eine Folge zu sehen, die einen Zyklus beinhaltet.

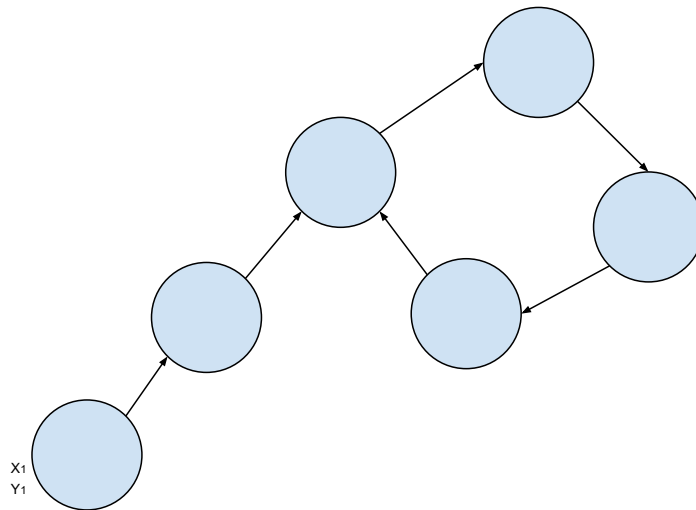


Abbildung 1: Schritt 1

Der Hase, hier dargestellt durch Y und der Igel, hier dargestellt durch X starten an der gleichen Position. Der Igel bewegt sich um ein Element, der Hase um zwei Elemente fort.

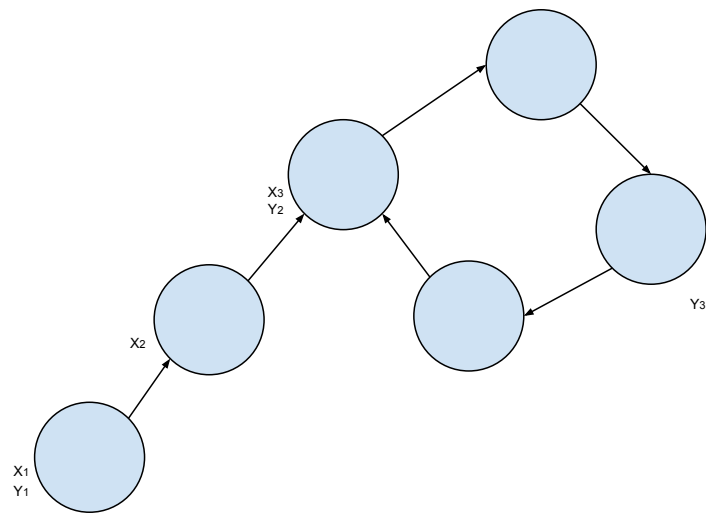


Abbildung 2: Schritt 2

Das ist der Zustand, bevor der Hase das erste mal im Kreis läuft. Es ist zu erkennen, dass der Hase doppelt so weit ist, wie der Igel.

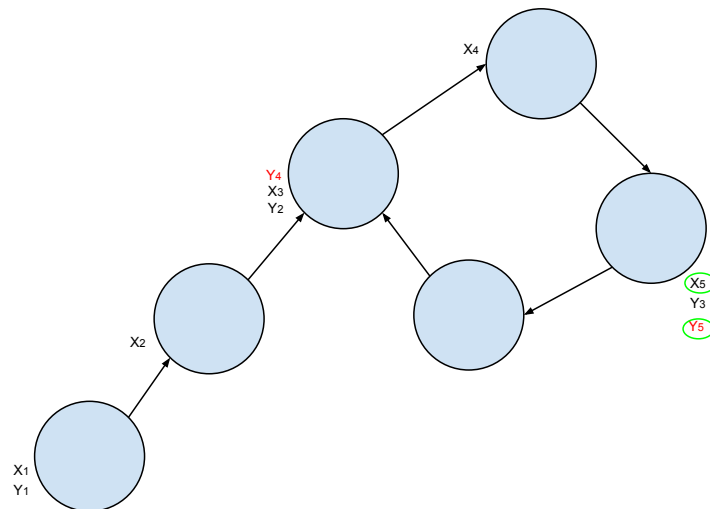


Abbildung 3: Schritt 3

Dies ist der Zustand, wenn Hase und Igel sich treffen. Die roten Zahlen zeigen, dass der Hase im Kreis läuft und die grün eingekreisten Zahlen zeigen, dass Hase und Igel sich in ihrem fünften Schritt treffen.

3.2.2 Mathematik

- Sei M eine endliche Menge mit der Abbildung $f : M \rightarrow M$. Die Menge M wird im Beispiel durch die Kreise dargestellt, die Abbildung f durch die Pfeile, die die Kreise verbinden.
- Man wähle $x_0 \in M$ und erzeuge die Folge x_0, x_1, x_2, \dots mit $x_{i+1} = f(x_i)$. Dies wird im Beispiel durch den Igel dargestellt, der sich entlang der Pfeile mit einer Schrittweite von 1 durch die Menge bewegt.
- $\exists i, j \in \mathbb{N}$, sodass $i \neq j$ und $x_i = x_j$ gilt. Hiermit wird gesagt, dass es einen Zyklus gibt. Wenn der Igel zwei mal auf das gleiche Feld kommt, muss er im Kreis gegangen sein.
- Die Folge y_0, y_1, y_2, \dots gegeben durch $y_0 = x_0$ und $y_{i+1} = f(f(y_i))$ ist gleich der Folge x_0, x_2, x_4, \dots . Hiermit wird der Hase beschrieben, der sich doppelt so schnell bewegt wie der Igel.
- Es gibt ein $c > 0$, sodass $x_c = x_{2c}$. Dies ist die Bedingung dafür, dass es möglich ist, dass sich Hase und Igel wieder treffen und zwar wenn beide gleich viele Schritte gemacht haben. c für den Igel, $2c$ für den Hasen, da der Hase eine doppelt so große Schrittweite hat wie der Igel.

Ein Beweis für diesen Algorithmus wird in Abschnitt 3.3.4 geführt.

3.3 Pollard Rho Algorithmus

3.3.1 Kongruenz modulo p

$a = px + r, b = py + r \mid a \equiv b \pmod{p} \iff px + r \equiv py + r \pmod{p} \iff px \equiv py \pmod{p} \iff p \mid (x - y)p \iff p \mid (x - y)$

3.3.2 Idee

3.3.3 Beispiel

3.3.4 Mathematik

3.4 Komplexität

4 Fazit

Literatur

[Lauritzen] *Niels Lauritzen. Concrete Abstract Algebra. Reprinted with corrections 2006*

[bk2boint] *blub*