Primfaktorzerlegung und Primzahltests

Maximilian Scholz

Proseminar Mathematik

June 24, 2014

Inhalt

Primzahlen

- Def.: Natürliche Zahlen > 1 die nur durch sich selbst und 1 teilbar sind.
- Es gibt unendlich viele Primzahlen. (Euklid)
- Jede natürliche Zahl lässt sich als Produkt von Primzalen darstellen. Bis auf die Reihenfolge ist diese Darstellung eindeutig. (Euklid)

Sieb des Eratosthenes

Sieb des Eratosthenes

- Der kleinste Primfaktor > 1 einer zusammengesetzten Zahl n ist eine Primzahl p.

Sieb des Eratosthenes

- Der kleinste Primfaktor > 1 einer zusammengesetzten Zahl n ist eine Primzahl p.
- Da p der kleinste Primfaktor ist, gilt $p \leq \frac{n}{p}$, also $p \leq \sqrt{n}$.

Sieb des Eratosthenes

- Der kleinste Primfaktor > 1 einer zusammengesetzten Zahl n ist eine Primzahl p.
- Da p der kleinste Primfaktor ist, gilt $p \leq \frac{n}{p}$, also $p \leq \sqrt{n}$.
- Alle zusammengesetzten Zahlen $n \leq N$ werden also beim Sieben mit einer Siebzahl q mit $q \leq \sqrt{n}$ gestrichen.
  $\Rightarrow$ Abbruchkriterium

Sieb des Eratosthenes

- Der kleinste Primfaktor > 1 einer zusammengesetzten Zahl n ist eine Primzahl p.
- Da p der kleinste Primfaktor ist, gilt $p \leq \frac{n}{p}$, also $p \leq \sqrt{n}$.
- Alle zusammengesetzten Zahlen $n \leq N$ werden also beim Sieben mit einer Siebzahl q mit $q \leq \sqrt{n}$ gestrichen.
  $\Rightarrow$ Abbruchkriterium
- Die übrigen Zahlen sind also Primzahlen.

Gebutstagsproblem

Gebutstagsproblem

- N Personen sind auf einem Geburtstag. Wie hoch ist die Wahrscheinlichkeit, dass zwei am gleichen Tag Geburtstag haben?

Gebutstagsproblem

- N Personen sind auf einem Geburtstag. Wie hoch ist die Wahrscheinlichkeit, dass zwei am gleichen Tag Geburtstag haben?
- Inverse Problem:
  Wie hoch ist die Wahrscheinlichkeit $P(N)$, dass kein Geburtstag mehrfach vorkommt?

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

Gebutstagsproblem

- N Personen sind auf einem Geburtstag. Wie hoch ist die Wahrscheinlichkeit, dass zwei am gleichen Tag Geburtstag haben?

- Inverse Problem:
  Wie hoch ist die Wahrscheinlichkeit $P(N)$, dass kein Geburtstag mehrfach vorkommt?

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

- Im Allgemeinen: $P(N) = \frac{365 \cdot 364 \ldots (365-N+1)}{365^N}$

  Für große N liegt die Zahl der Personen die man durchschnittlich braucht um eine Wiederholung zu erhalten bei $\sqrt{\frac{\pi N}{2}}$

Hase Igel Algorithmus

Hase Igel Algorithmus

- Sei M eine endliche Menge mit der Abbildung $f : M \to M$.

Hase Igel Algorithmus

- Sei M eine endliche Menge mit der Abbildung $f : M \rightarrow M$.
- Man wähle $x_0 \in M$ und erzeuge die Folge $x_0, x_1, x_2, \ldots$ mit $x_{i+1} = f(x_i)$.

Hase Igel Algorithmus

- Sei M eine endliche Menge mit der Abbildung $f : M \to M$.
- Man wähle $x_0 \in M$ und erzeuge die Folge $x_0, x_1, x_2, \ldots$ mit $x_{i+1} = f(x_i)$.
- $\exists\ i, j \in \mathbb{N}$, sodass $i \neq j$ und $x_i = x_j$ gilt.

Hase Igel Algorithmus

- Sei M eine endliche Menge mit der Abbildung $f : M \to M$.
- Man wähle $x_0 \in M$ und erzeuge die Folge $x_0, x_1, x_2, \ldots$ mit $x_{i+1} = f(x_i)$.
- $\exists\ i, j \in \mathbb{N}$, sodass $i \neq j$ und $x_i = x_j$ gilt.
- Die Folge $y_0, y_1, y_2, \ldots$ gegeben durch $y_0 = x_0$ und $y_{i+1} = f(f(y_i))$ ist gleich der Folge $x_0, x_2, x_4, \ldots$.

Kongruenz modulo p

- $a \equiv b \pmod{p} \Leftrightarrow p | (a - b)$

Kongruenz modulo p

- $a \equiv b \pmod{p} \Leftrightarrow p | (a - b)$
- $a = p \cdot x + r, \quad b = p \cdot y + r$

Kongruenz modulo p

- $a \equiv b \pmod{p} \Leftrightarrow p|(a-b)$
- $a = p \cdot x + r, \quad b = p \cdot y + r$
- $a - b = p(x-y) + (r-r) = p(x-y)$

Kongruenz modulo p

- $a \equiv b \pmod{p} \Leftrightarrow p|(a-b)$
- $a = p \cdot x + r, \quad b = p \cdot y + r$
- $a - b = p(x-y) + (r-r) = p(x-y)$
- $p|p(x-y)$

Fragen?

Pollard Rho Methode

Pollard Rho Methode

- Sei n eine zusammengesetzte Zahl und p ein Primfaktor von n.

Pollard Rho Methode

- Sei n eine zusammengesetzte Zahl und p ein Primfaktor von n.
- Gesucht sind a, b sodass
  $a \equiv b \pmod{p} \Rightarrow p | a - b$
  $a \not\equiv b \pmod{n} \Rightarrow n \nmid a - b$

Pollard Rho Methode

- Sei n eine zusammengesetzte Zahl und
  p ein Primfaktor von n.
- Gesucht sind a, b sodass
  $a \equiv b \pmod{p} \Rightarrow p | a - b$
  $a \not\equiv b \pmod{n} \Rightarrow n \not| a - b$
- Daraus folgt $1 < ggT(a - b, n) < n$.
  Wenn $a \neq b$ gilt, ist $ggT(a - b, n)$ ein nichttrivialer Primfaktor von n.

Pollard Rho Algorithmus

Pollard Rho Algorithmus

- Sei $f(x)$ eine ganzzahlige Polynomfunktion und $s \in \mathbb{Z}$.

Pollard Rho Algorithmus

- Sei $f(x)$ eine ganzzahlige Polynomfunktion und $s \in \mathbb{Z}$.
- Man erzeuge eine Folge von Pseudozufallszahlen mit:
  $x_0 = s,\ x_{i+1} = f(x_i) \mod n$.

Pollard Rho Algorithmus

- Sei $f(x)$ eine ganzzahlige Polynomfunktion und $s \in \mathbb{Z}$.
- Man erzeuge eine Folge von Pseudozufallszahlen mit: $x_0 = s,\ x_{i+1} = f(x_i) \mod n$.
- Wird schließlich periodisch, da beschränkt.

Pollard Rho Algorithmus

- Sei $f(x)$ eine ganzzahlige Polynomfunktion und $s \in \mathbb{Z}$.
- Man erzeuge eine Folge von Pseudozufallszahlen mit:
  $x_0 = s, \; x_{i+1} = f(x_i) \mod n$.
- Wird schließlich periodisch, da beschränkt.
- Anstatt $x_k \stackrel{?}{=} y_k$ suchen wir nach $1 \stackrel{?}{<} ggT(x_k - y_k, n) \stackrel{?}{<} n$

Beweis Teil 1

$\exists\ i, j \in \mathbb{N}$, sodass $i \neq j$ und $x_i = x_j$ gilt.

Beweis Teil 1

$\exists\ i, j \in \mathbb{N}$, sodass $i \neq j$ und $x_i = x_j$ gilt.

- $g : \mathbb{N} \to M$ gegeben durch $g(t) = f^t(x_0)$

Beweis Teil 1

$\exists\ i, j \in \mathbb{N}$, sodass $i \neq j$ und $x_i = x_j$ gilt.

- $g : \mathbb{N} \rightarrow M$ gegeben durch $g(t) = f^t(x_0)$
- M ist beschränkt also kann g nicht injektiv sein. Daraus folgt: $\exists\ i, j \in \mathbb{N}, i \neq j$, sodass $g(i) = g(j)$ und damit $x_i = x_j$ bei $i \neq j$.

Beweis Teil 2

Die Folge $y_0, y_1, y_2, ...$ gegeben durch $y_0 = x_0$ und $y_i + 1 = f(f(y_i))$ ist gleich der Folge $x_0, x_2, x_4, ...$

Beweis Teil 2

Die Folge $y_0, y_1, y_2, ...$ gegeben durch $y_0 = x_0$ und $y_i + 1 = f(f(y_i))$ ist gleich der Folge $x_0, x_2, x_4, ...$

- Angenommen $x_i = x_j$ für $j > i$.

Beweis Teil 2

Die Folge $y_0, y_1, y_2, \ldots$ gegeben durch $y_0 = x_0$ und $y_i + 1 = f(f(y_i))$ ist gleich der Folge $x_0, x_2, x_4, \ldots$

- Angenommen $x_i = x_j$ für $j > i$.
- Falls $c \geq i$ und $2c = c + k(j - i) \geq i$ mit $k \geq 0$ muss $x_c = x_{2c}$ gelten.

Beweis Teil 2

Die Folge $y_0, y_1, y_2, \ldots$ gegeben durch $y_0 = x_0$ und $y_i + 1 = f(f(y_i))$ ist gleich der Folge $x_0, x_2, x_4, \ldots$

- Angenommen $x_i = x_j$ für $j > i$.
- Falls $c \geq i$ und $2c = c + k(j - i) \geq i$ mit $k \geq 0$ muss $x_c = x_{2c}$ gelten.
- Man wähle $k \geq 0$, sodass $c = k(j - 1) \geq i$ und erhält das gesuchte c.

Beweis Teil 2

Die Folge $y_0, y_1, y_2, ...$ gegeben durch $y_0 = x_0$ und $y_i + 1 = f(f(y_i))$ ist gleich der Folge $x_0, x_2, x_4, ...$

- Angenommen $x_i = x_j$ für $j > i$.
- Falls $c \geq i$ und $2c = c + k(j - i) \geq i$ mit $k \geq 0$ muss $x_c = x_{2c}$ gelten.
- Man wähle $k \geq 0$, sodass $c = k(j - 1) \geq i$ und erhält das gesuchte c.
- Aus $x_{m+2} = f(f(x_m))$ folgt $y_m = x_{2m}$.

Pollard Rho Beispiel

Gesucht: Primfaktorzerlegung von N=143

Parameter: $x_0 = y_0 = 0,\ f(x) = (x^2 + 1) \mod N$

k	$x_k = f(x_{k-1})$	$y_k = f(f(y_{k-1}))$	$ggT(x_k - y_k, N)$
0	0	0	0
1	1	2	1
2	2	26	1
3	5	15	1
4	26	26	143
5	105	15	1
6	15	26	11

Pollard Rho Beispiel

Gesucht: Primfaktorzerlegung von N=143

Parameter: $x_0 = y_0 = 0,\ f(x) = (x^2 + 1) \mod N$

k	$x_k = f(x_{k-1})$	$y_k = f(f(y_{k-1}))$	$ggT(x_k - y_k, N)$
0	0	0	0
1	1	2	1
2	2	26	1
3	5	15	1
4	26	26	143
5	105	15	1
6	15	26	11

- Mit $\frac{143}{11} = 13$ erhält man den zweiten Primfaktor.

Pollard Rho Komplexität

- Wir suchen keine Geburtstag aber Wiederholungen $\pmod{p}$.

  $x_k \equiv x_{2k} \pmod{p}$
- Wir erhalten mithilfe des Geburtstagsproblemes $\mathcal{O}(\sqrt{\frac{\pi p}{2}})$
- Da $p \leq \sqrt{n}$ gilt $\sqrt{p} \leq \sqrt[4]{n}$
  $\Rightarrow \mathcal{O}(\sqrt[4]{n})$
- Fazit: Nach durchschnittlich $\sqrt[4]{n}$ Versuchen findet man einen Primfaktor von n.
- Wichtig ist noch die Geschwindigkeit des ggT $\rightarrow$ Euklid und der Aufwand der Funktion f.
- Zusammen ergibt dies $(\mathcal{O}(\text{Euklid}) + 3\mathcal{O}(f)) \cdot \mathcal{O}(\sqrt[4]{n})$

Quellen

- Niels Lauritzen. Concrete Abstract Algebra. Reptrinted with corrections 2006
- `www.bk2boint.dnsalias.org/int_neu/tl_files/Material%20Informatik/erathostenes/sieb.pdf`