

Relationen und Abbildungen

Relationen

Was genau ist eine Relation?

Relationen

Was genau ist eine Relation?

Antwort: „Eine Beziehung zwischen zwei Dingen“

Relationen

Beispiel:

Eine der bekanntesten Relationen ist \leq (kleiner gleich)

Zwei Zahlen a und b stehen nun dann in Relation \leq , wenn $a \leq b$ gilt

So steht etwa 3 in Relation zu 5, da $3 \leq 5$ gilt.

Andererseits gilt nicht $5 \leq 3$, dass heißt 5 ist nicht in Relation zu 3.

Es gibt viele Beispiele für Relationen und viele davon kennen wir schon.

Etwa:

\leq , \neq , $<$, $=$

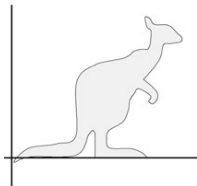
Relationen

Mathematisch definiert:

Definition:

Seien M und N Mengen und $R \subset M \times N$. Dann heißt R **Relation** auf $M \times N$. Gilt $M = N$, dann ist R *Relation* auf M .

Das heißt R ist eine Teilmenge des kartesischen Produkts von M und N .
 R ist beliebig definierbar, dh man könnte auch eine Relation definieren, die so aussieht:



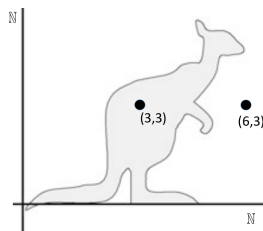
Relationen

Anmerkung: Eine Relation kann für ein Paar von Elementen nur wahr oder falsch sein! Entweder etwas steht in Relation zu etwas anderem oder nicht.

Anmerkung: Eine Relation gilt immer nur zwischen zwei Elementen!

Anmerkung: $3 \leq 5$ ist die Kurzschreibweise für $R_{\leq}(3, 5)$. Alternativ kann man auch $3R5$ oder $(3, 5) \in R$ schreiben.

Relationen



Dieses Bild definiert eine Relation auf den Mengen $\mathbb{N} \times \mathbb{N}$. Nehmen wir den eingezeichneten Punkt $(3,3)$. Er liegt im Känguru, dh es gilt $R(3,3)$. Die Relation von 3 zu 3 ist wahr. Dagegen gilt nicht $R(6,3)$.

Alles was eine Relation macht, ist eine Aussagevorschrift bezüglich dem Verhältnis zweier Elemente.

Relationen

Beispiel:

Wir behaupten $<$ ist eine Relation. Warum stimmt das?

Beweisskizze:

- $<$ auf der Menge $\mathbb{N} \times \mathbb{N}$ definiert
- für alle Zahlen $a, b \in \mathbb{N}$ kann man entscheiden, ob $R_{<}(a, b)$ wahr oder falsch ist. Wir wissen, dass $2 < 5$ gilt, also ist $R_{<}(2, 5) = 1$. Da wir dies für absolut alle Zahlen a, b entscheiden können, können wir behaupten, dass $<$ eine Relation ist.

Relationen

Wir haben bis jetzt 2-stellige Relationen gehabt, aber man kann auch n -stellige Relationen haben.

Definition:

Seien N_1, N_2, \dots, N_k Mengen und $R \subset N_1 \times N_2 \times \dots \times N_k$. Dann heißt R Relation auf $N_1 \times N_2 \times \dots \times N_k$.

Relationen

Im Prinzip funktioniert eine n -stellige Relation genau gleich, wie eine 2-stellige.

Beispiel:

Wir haben $R_{<<}(a, b, c)$, also die Relation, die bestimmt, ob $a < b < c$ hält. $R_{<<}(1,2,3)$ etwa wäre wahr. Hingegen ist $R_{<<}(3, 1, 2)$ falsch. Wir sehen, dass n -stellige Relationen so funktionieren, wie wir es erwarten.

Anwendung: Relationales Datenmodell

Relationales Datenmodell

Relationale Datenbanken repräsentieren Relationen in Tabellen.

- **Zeile:** Tupel, d.h. ein Element aus der n -stelligen Relation
- **Spalte:** Attribut, d.h. Stelle (Dimension) der n -stelligen Relation

Tabelle (Relation) **Produkt:**

PNr	Typ	Preis	HNr
1	Laptop	990	1
2	PC	200	2
3	Server	100	3

Relationales Datenmodell

- Jede Zeile in der Tabelle **Produkt** ist ein Element (auch n-Tupel genannt) der n-stellige Relationen die das kartesische Produkt aus den Mengen $\mathbb{N} \times \text{char}(20) \times \mathbb{N} \times \mathbb{N}$,
- Die Datenbankinhalte der Tabelle **Hersteller** auf der folgenden Seite bestehen aus Elementen der kartesische Produktmenge $\mathbb{N} \times \text{char}(20) \times \text{char}(20)$.

Relationales Datenmodell

Hersteller:

HNr	Name	Ort
1	IBM	New York
2	Apple	Cupertino
3	HP	Paolo Alto

Relationales Datenmodell

- Relationale Datenbanken bestehen oft aus sehr vielen Tabellen (tausende).
- Die relationale Algebra bietet Operationen, die das Arbeiten mit Tabellen erleichtert,
- implementiert in der Structured Query Language (SQL).

Wichtige Operationen

Selektion:

eine Teilmenge der vorhandenen Zeilen wird ausgewählt. z.B. ein bestimmter Hersteller aus der Tabelle der Hersteller.

SQL: `SELECT * FROM Hersteller WHERE Name = 'IBM'`

Hersteller

HNr	Name	Ort
1	IBM	New York
2	Apple	Cupertino
3	HP	Paolo Alto

Selektionsergebnis

HNr	Name	Ort
1	IBM	New York

Wichtige Operationen

Projektion:

eine Teilmenge der vorhandenen Spalten wird ausgewählt. z.B. die Spalten Typ und Preis der Tabelle Produkt.

SQL: SELECT Typ, Preis FROM Produkt

Produkt

PNr	Typ	Preis	HNr
1	Laptop	990	1
2	PC	200	2
3	Server	100	3

Projektionsergebnis

Typ	Preis
Laptop	990
PC	200
Server	100

Wichtige Operationen

Join:

Verbund von Relationen; dies geschieht unter Heranziehung gemeinsamer Attribute (Spalten). Beispiel: Jede Zeile von **Hersteller** und **Produkt** wird anhand von übereinstimmender Werte in HNr aneinandergefügt.

SQL: `SELECT * FROM Produkt JOIN Hersteller ON Produkt.HNr = Hersteller.HNr`

PNr	Typ	Preis	HNr	Name	Ort
1	Laptop	900	1	IBM	New York
2	PC	200	2	Apple	Cupertino
3	Server	100	3	HP	Palo Alto

Äquivalenzrelationen

Äquivalenzrelationen

Definition:

Äquivalenzrelationen sind Relationen auf einer Menge M mit den folgenden Eigenschaften:

Reflexivität: $\forall x \in M$ gilt $R(x, x)$.

Symmetrie: $\forall x, y \in M$ mit $R(x, y)$ gilt $R(y, x)$

Transitivität: $\forall x, y, z \in M$ mit $R(x, y)$ und $R(y, z)$ gilt $R(x, z)$.

Äquivalenzrelationen

Definition:

Äquivalenzrelationen sind diejenigen Relationen, die die folgenden Eigenschaften haben:

Reflexivität: $\forall x \in M$ gilt $R(x, x)$.

Symmetrie: $\forall x, y \in M$ mit $R(x, y)$ gilt $R(y, x)$.

Transitivität: $\forall x, y, z \in M$ mit $R(x, y)$ und $R(y, z)$ gilt $R(x, z)$.

Was bedeutet das genau?

Äquivalenzrelationen

Reflexivität: $\forall x \in M$ gilt $R(x, x)$.

Gelesen wird der Satz folgendermaßen: „Für alle x aus M gilt, dass x in Relation zu x ist.“

Dies bedeutet, dass ein Element immer in Relation zu sich selbst steht. Nehmen wir als Relation \leq . Die Behauptung ist, dass jede Zahl immer in Relation zu sich selbst steht, also $x \leq x$ für alle $x \in \mathbb{R}$ gilt. $3 \leq 3$ gilt genauso wie $5 \leq 5$ und dies gilt auch für alle anderen Zahlen. \leq ist reflexiv.

Äquivalenzrelationen

Symmetrie: $\forall x, y \in M$ mit $R(x, y)$ gilt $R(y, x)$.

Gelesen wird der Satz folgendermaßen: „Für alle x und y aus M gilt, dass wenn x in Relation zu y ist, dann ist auch y in Relation zu x .“

Wenn ein Element in Relation zu einem anderen Element steht, dann gilt das auch umgekehrt. Nehmen wir als Relation etwa \neq her. So gilt natürlich wenn $x \neq y$ stimmt, dass auch $y \neq x$ stimmt.

Äquivalenzrelationen

Transitivität: $\forall x, y, z \in M$ mit $R(x, y)$ und $R(y, z)$ gilt $R(x, z)$.

Gelesen wird der Satz folgendermaßen: „Für alle x , y und z aus M gilt, dass wenn x in Relation zu y ist und y zu z dann ist auch x in Relation zu z .“

Nehmen wir als Relation $=$. Wenn wir wissen, dass $x = y$ ist und $y = z$, dann ist auch klar, dass $x = z$ gilt.

Äquivalenzrelationen

Beispiele:

Ist $R_<$ eine Äquivalenzrelation?

Wir wissen schon, dass $<$ eine Relation ist. Nun müssen wir nur noch die drei Äquivalenz-Eigenschaften überprüfen:

1) Gilt Reflexivität?

Nein, $x < x$ gilt nicht! Das heißt $R_<$ ist keine Äquivalenzrelation

Äquivalenzrelationen

Beispiele:

Was ist mit $R_=_$?

$x = x$ gilt natürlich für alle Zahlen \Rightarrow reflexiv

Wenn $x = y$ gilt, dann gilt auch $y = x \Rightarrow$ symmetrisch

Transitivität haben wir schon gezeigt

$\Rightarrow R_=_$ ist eine Äquivalenzrelation

Äquivalenzrelationen

Betrachten wir

$$R_5 := \{(m, n) \in \mathbb{Z} \times \mathbb{Z} \mid m - n \text{ ist ohne Rest durch 5 teilbar} \} \subset \mathbb{Z} \times \mathbb{Z}$$

Dass R_5 eine Relation ist, ergibt sich aus der Definition: Wir betrachten eine Teilmenge von $\mathbb{Z} \times \mathbb{Z}$.

Ist es auch eine Äquivalenzrelation?

Anmerkung: $n \in \mathbb{N}$ ist ohne Rest durch 5 teilbar bedeutet:

$$\exists k \in \mathbb{Z} : 5 \cdot k = n$$

Äquivalenzrelationen

Reflexivität: $R_5(n, n)$ gilt, denn:

- $n - n = 0$
- und 0 ist durch 5 teilbar ($0 \cdot 5 = 0$) \Rightarrow **Reflexivität**

Symmetrie: Zu zeigen: $R_5(n, m) \Rightarrow R_5(m, n)$

- Es gelte $R_5(n, m) \Rightarrow \exists k : n - m = 5 \cdot k$
- $\Rightarrow m - n = 5 \cdot (-k)$
- $\Rightarrow \exists k : m - n = 5 \cdot k \Rightarrow R_5(m, n) \Rightarrow$ **Symmetrie**

Äquivalenzrelationen

Transitivität: Zu zeigen: $R_5(n, m)$ und $R_5(m, s) \Rightarrow R_5(n, s)$

- Es gelte $R_5(n, m)$ und $R_5(m, s)$
- $\Rightarrow \exists k : n - m = 5 \cdot k$ und $\exists l : m - s = 5 \cdot l$
- $\Rightarrow n - s = (n - m) + (m - s) = 5 \cdot k + 5 \cdot l = 5 \cdot (k + l)$
- $\Rightarrow n - s$ ist durch 5 teilbar, dh $R_5(n, s)$ gilt \Rightarrow **Transitivität**

Äquivalenzrelationen

Definition:

Sei R eine Äquivalenzrelation auf M und $a \in M$. Dann heißt die Menge $[a] := \{x \in M \mid R(x, a)\}$

Äquivalenzklasse von a .

Dh alle jene Elemente, die in Relation zu a stehen.

Äquivalenzrelationen

Beispiel:

Wir haben R_5 als Äquivalenzrelation. Dann sind die Äquivalenzklassen, diejenigen Zahlen die bei der Division durch 5 den gleichen Rest haben.

zB $[1] := \{1, 6, 11, 16, 21, \dots\}$

Denn für jede dieser Zahlen gilt $R(x, 1)$.

Beweis: Zahlen der Form $x \in \{1, 6, 11, 16, 21, \dots\}$ kann man darstellen als $\exists k : x = 5 \cdot k + 1$. Das können wir umformen zu $\exists k : 5 \cdot k = x - 1$ und das ist die Definition von $R(x, 1)$.

Ordnungsrelationen

Ordnungsrelationen

Ordnungsrelationen sind eine bestimmte Klasse von Relationen.

Die (für uns) wichtigen Ordnungsrelationen kennen wir alle schon:
Es sind: \leq , $<$.

Anmerkung: \geq und $>$ sind eigentlich die gleichen Relationen wie \leq und $<$. Sie sind nur umgekehrt definiert.

Ordnungsrelationen

Definition:

Ordnungsrelationen sind diejenigen Relationen, die die folgenden Eigenschaften haben:

Reflexivität: $\forall x \in M$ gilt $R(x, x)$.

Anti-Symmetrie: $\forall x, y \in M$ mit $R(x, y)$ und $R(y, x)$ gilt $x = y$.

Transitivität: $\forall x, y, z \in M$ mit $R(x, y)$ und $R(y, z)$ gilt $R(x, z)$.

Genau genommen ist $<$ keine Ordnungsrelation, warum? Nicht reflexiv! Häufig bezeichnet als *strikte Ordnungsrelation* oder *Halbordnung*. Aber eine Sache der Definition. Man sollte sich immer bewusst sein welche Definitionen zugrunde liegen.

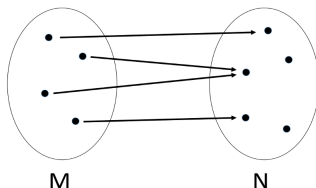
Abbildungen

Abbildungen

Definition:

Seien M und N Mengen. Jedem $x \in M$ wird genau ein $y \in N$ zugeordnet. Diese Zuordnung definiert eine **Abbildung** von M nach N .

Im Grunde genommen ist das ganz einfach. Man sagt jedem Element aus einer Menge, welches andere Element zu ihm dazugehört.



Abbildungen

Schreibweise:

Meistens wird eine Abbildung mit einem Kleinbuchstaben wie f oder g bezeichnet.

Das Element auf das ein $x \in M$ abgebildet wird, wird dann als $f(x)$ bezeichnet.

Insgesamt wird das dann folgendermaßen bezeichnet:

$$f : M \rightarrow N, x \mapsto f(x)$$

Der Pfeil \rightarrow wird für Mengen verwendet, \mapsto für die einzelnen Elemente.

Abbildungen

Definitionen:

$D(f) := M$ ist die **Definitionsmenge** von f

$x \in M$ ist das **Argument** von f

$f(M) = \{y \in N \mid \text{es gibt ein } x \in M : y = f(x)\}$ ist die **Bildmenge** von f

Sei $x \in M, y \in N$ und $y = f(x)$,
dann ist y **Bild von x** und x **Urbild von y** .

In der Mathematik muss man sehr genau sein und exakt wissen von was gesprochen wird, deswegen sind all diese Definitionen notwendig!

Abbildungen

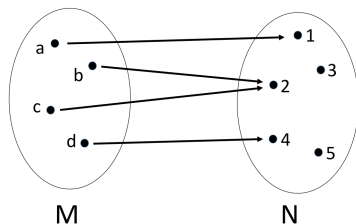
Definitionen:

Ist $U \subset M$, dann ist die Menge der Bilder von $x \in U$ das **Bild von U**. Dies wird mit $f(U) := \{f(x) | x \in U\}$ bezeichnet.

Ist $V \subset N$, dann ist die Menge der Urbilder von $y \in V$ das **Urbild von V**. Dies wird mit $f^{-1}(V) := \{x \in M | f(x) \in V\}$ bezeichnet.

Abbildungen

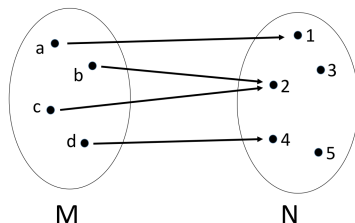
Wir betrachten diese Definitionen anhand eines Beispiels:



Die **Definitionsmenge** ist schlicht M selbst, also die Menge $\{a, b, c, d\}$. Die **Bildmenge** von f sind diejenige Elemente, die „getroffen“ werden, also $\{1, 2, 4\}$. Betrachten wir a . Das **Bild** von a ist 1 . Das **Urbild** von 1 ist a .

Abbildungen

Und nun das Ganze für Mengen:



Nehmen wir an wir betrachten die Menge $\{a, b\}$. Das **Bild** davon ist dann $\{1, 2\}$. Allerdings: Das **Urbild** von $\{1, 2\}$ ist $\{a, b, c\}$.

Abbildungen

Definitionen:

Sei $f : M \rightarrow N$ eine Abbildung. Dann heit f :

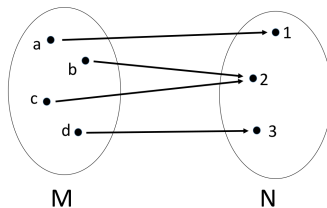
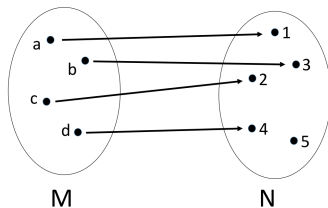
Injektiv: fr alle $x_1, x_2 \in M$ mit $x_1 \neq x_2$ gilt: $f(x_1) \neq f(x_2)$

Surjektiv: fr alle $y \in N$ gibt es ein $x \in M$ mit $f(x) = y$

Bijektiv: wenn f injektiv und surjektiv ist

Abbildungen

Beispiele:



Sind diese Beispiele injektiv/surjektiv/bijektiv?

Abbildungen

Die **Mächtigkeit** einer Menge sagt aus, wie groß eine Menge ist. Bei endlichen Mengen ist das leicht, man zählt einfach die Elemente ab. So hat etwa $M = \{a, b, c, d\}$ die Mächtigkeit $|M| = 4$.

Zwei Mengen M und N haben dieselbe Mächtigkeit, wenn es eine bijektive Abbildung $f : M \rightarrow N$ zwischen den Mengen gibt. Man sagt sie haben dieselbe **Kardinalität**.

Unendlich große Mengen, wie \mathbb{N} und \mathbb{R} , haben keine endlich Kardinalität. Man sagt, dass \mathbb{N} (und jede dazu bijektive Menge) **abzählbar unendlich** ist. \mathbb{R} hingegen ist **überabzählbar unendlich**.

Abbildungen

Beispiel:

Welche Kardinalität hat \mathbb{Z} ?

Wir betrachte die Abbildung $f : \mathbb{N}_0 \rightarrow \mathbb{Z}$: $0 \mapsto 0, 1 \mapsto 1, 2 \mapsto -1, 3 \mapsto 2, 4 \mapsto -2, 5 \mapsto 3, 6 \mapsto -3$ usw.

Wir sehen: f ist bijektiv $\Rightarrow \mathbb{Z}$ und \mathbb{N} haben dieselbe Mächtigkeit.

$\Rightarrow \mathbb{Z}$ und \mathbb{N} sind also gleich groß, auch wenn \mathbb{N} eine echte Teilmenge von \mathbb{Z} ist.

Anwendung: Hashing

Hashfunktionen

Abbildungen die Datensätzen beliebiger Länge (viele Bit) **Hashwerte** fester, meist kürzerer Länge (z.B. 128 Bit) zuordnen.
Dies sind dann Speicheradressen welche effizientes Suchen von Datensätzen in grossen Datenmengen ermöglichen.

Beispiel Hashing

Aufgabe: Finde schnell die Vorwahl von einem beliebigen Ort in Österreich.

Schlüssel (key): Suchbegriff ist in unserem Fall der Ort, mithilfe dessen die Speicheradresse berechnet wird.

Wert (value): gesuchter Wert ist in unserem Fall die Vorwahl.

Idee: Nachdem wir die Speicheradresse mithilfe des Schlüssels berechnen, ist der Zugriff über den Suchbegriff möglich. Wir müssen also nicht mehr die gesamte Datenmenge durchsuchen, sondern können nur die Werte betrachten, welche an der berechneten Speicheradresse zu finden sind, da wir immer (key, value) - Paare abspeichern.

Die Hashfunktion

Die **Speicheradresse** die mit Hilfe des Suchbegriffes (sprich Schlüssel k) berechnet wird, wird mittels Hashfunktion (H) ermittelt welche k als Argument erhält. Der Schlüssel wird also unter der Adresse $H(k)$ abgelegt und kann auch damit wiedergefunden werden.

$$\begin{aligned} H : K &\rightarrow A = \{0, 1, 2, \dots, N - 1\} \\ k &\mapsto H(k) \end{aligned}$$

Beispiel Hashfunktion

Die Hashfunktion:

$$H(k) = \sum_i a_i \mod N \quad (1)$$

a_i = die Zahl für die Stelle des i -ten Buchstaben z.b. $k = XYZ$ würde bedeuten $a_1 = 24$, $a_2 = 25$, $a_3 = 26$.

$N = 7$ Wenn wir den Wert fuer die Schluessel WIEN, GRAZ, SALZBURG und DORNBIRN berechnen.

Wir erhalten:

$$H(WIEN) = (23 + 9 + 5 + 14) \mod 7 = 2$$

$$H(GRAZ) = ?$$

$$H(SALZBURG) = ?$$

$$H(DORNBIRN) = ?$$

Welche Adressen wuerden fuer GRAZ, SALZBURG und DORNBIRN mit der gegebenen Hashfunktion berechnet?

Beispiel Hashfunktion

Die Hashfunktion:

$$H(k) = \sum_i a_i \mod N$$

a_i = die Zahl für die Stelle des i -ten Buchstaben z.b. $k = XYZ$ wuerde bedeuten $a_1 = 24$, $a_2 = 25$, $a_3 = 26$.

$N = 7$ Wenn wir den Wert für die Schlüssel WIEN, GRAZ, SALZBURG und DORNBIRN berechnen.

Wir erhalten:

$$H(WIEN) = (23 + 9 + 5 + 14) \mod 7 = 2$$

$$H(GRAZ) = (7 + 18 + 1 + 26) \mod 7 = 3$$

$$H(SALZBURG) = (19 + 1 + 12 + 26 + 2 + 21 + 18 + 7) \mod 7 = 1$$

$$H(DORNBIRN) = (4 + 15 + 18 + 14 + 2 + 9 + 18 + 14) \mod 7 = 3$$

Kollision

Wir sehen das wir ein Problem haben, da sowohl Dornbirn als auch Graz **dem selben Speicherplatz zugeordnet** werden, ein Sachverhalt der Kollision genannt wird.

Normalerweise gibt es zu viele Schlüssel, die auf eine kleinere Anzahl verfügbarer Hashwerte (sprich Speicheradressen) mittels $H(k)$ abgebildet werden sollen.

Kollision

Eine einfache Strategie zur Auflösen von Kollisionen ist das **lineare Sondieren**. Dabei wird die nächst grössere freie Adresse gewählt.

$$s_i(k) = (H(k) + i) \mod N, \text{ für } i = 1, \dots, N - 1.$$

Hashtabelle

Eine **Hashtabelle** ist eine Datenstruktur in der die (key, value)- Paare an der mit der Hashfunktion berechneten Adresse abgespeichert werden. Wir vermeiden die Kollision zwischen GRAZ und DORNBIRN indem wir DORNBIRN an der Adresse $(3 + 1) \bmod 7 = 4$ abspeichern.

Speicherplatz(n)	Schlüssel(k_n)	Wert(v_n)
0		
1	SALZBURG	0662
2	WIEN	01
3	GRAZ	0316
4	DORNBIRN	05572
5		
6		

Quadratisches Sondieren

Bessere Strategie zur Auflösung von Kollisionen:

$$\left. \begin{array}{l} s_{2i-1}(k) = (H(k) + i^2) \mod N \\ s_{2i}(k) = (H(k) - i^2) \mod N \end{array} \right\} i = 1, \dots, \frac{N-1}{2}.$$

d.h. für $N = 7$ gilt:

$$H(k) + 1, H(k) - 1, H(k) + 4, H(k) - 4, H(k) + 9, H(k) - 9$$

Quadratisches Sondieren. Wie gut funktioniert es?

i	1		2		3	
i^2	1		4		9	
$\pm i^2 \bmod 7$	1	6	4	3	2	5

i	1		2	
i^2	1		4	
$\pm i^2 \bmod 5$	1	4	4	1

Abbildung: Adressen für verschiedenes N .

Satz von Fermat

Ist N eine Primzahl und es gilt $N \bmod 4 = 3$, so gilt

$$\{\pm i^2 \bmod N : i = 1, \dots, \frac{N-1}{2}\} = \{1, 2, \dots, N-1\}.$$

Wähle N gemäß dieser Vorschrift, damit eine gleichmässige Auslastung der Hashtabelle auch im Fall von Kollisionen gewährleistet ist.