Division mit Rest — Theorie

Auf- und Abrundefunktion

- $\lfloor x \rfloor$ ist die grösste ganze Zahl, die kleiner oder gleich x ist.
- $\lceil x \rceil$ ist die kleinste ganze Zahl, die grösser oder gleich x ist.

Beispiele:

(a) $\lfloor 1.7 \rfloor =$

(b) $\lceil 1.7 \rceil =$

(c) $\lfloor -1.7 \rfloor =$

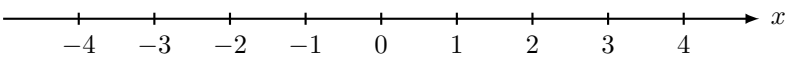
(d) $\lceil -1.7 \rceil =$

(e) $\lfloor 3 \rfloor =$

(f) $\lceil 3 \rceil =$

(g) $\lfloor -3 \rfloor =$

(h) $\lceil -3 \rceil =$



`floor(...)` und `ceil(...)`

In Python heissen die Auf- und Abrundefunktion `ceil(x)` und `floor(x)`, wobei `x` eine Zahl sein muss. Um diese Funktionen benutzen zu können muss man sie vor dem ersten Gebrauch importieren.

```
1  from math import floor, ceil
2
3  print(floor(7.7))   # => ?
4
5  print(ceil(7.7))    # => ?
6
7  print(floor(16))    # => ?
8
9  print(ceil(-27))    # => ?
10
11 print(floor(-8.7))  # => ?
12
13 print(ceil(-3.1))   # => ?
```

Die Ganzzahldivision

Wir können die Abrundefunktion dazu benutzen, die Ganzahldivsion (*integer division*) zu definieren. Sind a und b ganze Zahlen, so bezeichnet

$$\left\lfloor \frac{a}{b} \right\rfloor,$$

wie oft der Divisor b im Dividenden a enthalten ist.

Beispiele:

(a) $\left\lfloor \frac{7}{3} \right\rfloor =$

(b) $\left\lfloor \frac{513}{10} \right\rfloor =$

(c) $\left\lfloor \frac{19}{17} \right\rfloor =$

(d) $\left\lfloor \frac{20}{5} \right\rfloor =$

Der Python-Operator //

In Python kann eine Ganzzahldivision mit dem Operator `//` ausgeführt werden

```
1  print(18 // 5)    # ?
2
3  print(27 // 2 )   # ?
4
5  print(125 // 20)  # ?
6
7  print(-12 // 3)   # ?
8
9  print(71 // -10)  # ?
10
11 print(100 // 3)   # ?
```

Der Divisionsrest

Sind a und b natürliche Zahlen und $b \neq 0$, so definieren wir den *Divisionsrest* $a \bmod b$ [sprich: „a modulo b“] wie folgt:

$$a \bmod b = a - \left\lfloor \frac{a}{b} \right\rfloor \cdot b$$

Beispiele

(a) $7 \bmod 3 =$

(b) $14 \bmod 5 =$

(c) $9 \bmod 13 =$

(d) $18 \bmod 6 =$

(d) $0 \bmod 4 =$

Der Python-Operator %

In Python kann der Divsionsrest mit dem Operator `%` ausgeführt werden

```
1  print(18 % 5)    # ?
2
3  print(27 % 2)    # ?
4
5  print(125 % 20)  # ?
6
7  print(12 % 3)    # ?
8
9  print(71 % 10)   # ?
10
11 print(4 % 15)    # ?
12
13 print(-100 % 3)  # ?
```

Restklassen

Zwei Zahlen a und b gehören in die gleiche *Restklasse modulo* der Zahl n, wenn sie bei der Division durch n den gleichen Rest haben. Dividiert man alle Elemente der folgenden Mengen durch $n = 3$, erhält man jeweils denselben Rest.

Für alle $a \in \{\ldots, -6, -3, 0, 3, 6, \ldots\}$ ergibt $a : 3$ den Rest 0

Für alle $a \in \{\ldots, -5, -2, 1, 4, 7, \ldots\}$ ergibt $a : 3$ den Rest 1

Für alle $a \in \{\ldots, -4, -1, 2, 5, 8, \ldots\}$ ergibt $a : 3$ den Rest 2

Wir behandeln diese Mengen als neue Objekte und verwenden in der Regel die kleinste nichtnegative Zahl r darin, um die Restklasse modulo n mit $[r]_n$ zu bezeichnen.

$[0]_3 = \{\ldots, -6, -3, 0, 3, 6, \ldots\}$ Restklasse von 0 modulo 3

$[1]_3 = \{\ldots, -5, -2, 1, 4, 7, \ldots\}$ Restklasse von 1 modulo 3

$[2]_3 = \{\ldots, -4, -1, 2, 5, 8, \ldots\}$ Restklasse von 2 modulo 3

Wenn es klar ist, um welchen Teiler n es sich handelt, schreibt man statt $[r]_n$ auch kürzer $[r]$ oder nur $\overline{r}$.

Rechnen mit Restklassen

Damit definieren wir die Addition und Multiplikation von Restklassen:

$$[a]_n + [b]_n = [a + b]_n \quad \text{und} \quad [a]_n \cdot [b]_n = [a \cdot b]_n$$

wobei a und b beliebige Elemente (*Repräsentanten*) aus der Restklasse $[a]_n$ bzw, $[b]_n$ sind. Die Resultate hängen nicht von der Wahl der Repräsentanten ab. *Beispiele:*

(a) $[2]_3 + [1]_3 = [2 + 1]_3 = [3]_3 = [0]_3$

(b) $[8]_3 + [-2]_3 = [8 + (-2)]_3 = [6]_3 = [0]_3$

(c) $[2]_3 \cdot [2]_3 = [2 \cdot 2]_3 = [4]_3 = [1]_3$

(d) $[-4]_3 \cdot [-1]_3 = [(-4) \cdot (-1)]_3 = [4]_3 = [1]_3$

Additions- und Multiplikationstafeln

Für $n = 5$:

+	**0**	**1**	**2**	**3**	**4**
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

×	**0**	**1**	**2**	**3**	**4**
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

Um weniger zu schreiben, lassen wir die eckigen Klammern weg.

Subtraktion

Die Subtraktion von Restklassen ist analog zur Addition definiert: $[a]_n - [b]_n = [a-b]_n$.

Auch hier ist die Definition unabhängig von der Wahl der Repräsentanten, wie die folgenden Beispiele nahe legen.

(a) $[1]_5 - [4]_5 = [1-4]_5 = [-3]_5 = [2]_5$

(b) $[11]_5 - [9]_5 = [11-9]_5 = [2]_5$

Division

Die Division von Restklassen lässt sich nicht so einfach definieren. Wir ersetzen daher die Division durch die Multiplikation mit dem *multiplikativen inversen Element*, so wie das bei den rationalen Zahlen gemacht wird:

$3 : 5 = 3 \cdot 5^{-1} = 3 \cdot \frac{1}{5} = \frac{3}{5} \quad$ mit $5^{-1} \cdot 5 = \frac{1}{5} \cdot 5 = 1$

Ebenso gehen wir bei den Restklassen vor. Wenn wir durch $[b]_n$ dividieren wollen, suchen wir eine Restklasse $[b]_n^{-1}$ für die $[b]_n \cdot [b]_n^{-1} = [1]_n$ gilt.

Die multiplikative Inverse einer Restklasse, sofern sie existiert, kann mit dem *erweiterten euklidischen Algorithmus* bestimmt werden. Es genügt hier aber, die Inverse aus der Multiplikationstabelle abzulesen. Für $n = 5$ finden wir z. B. $[2]_5^{-1} = [3]_5$, denn $[2]_5 \cdot [3]_5 = [1]_5$.

Elemente ohne Inverse

Es gibt Restklassen ohne Inverse. $[0]_n$ ist immer eine solche Restklasse aber wenn n keine Primzahl ist, so gibt weitere Elemente, für die es keine Inverse gibt, wie etwa $[3]_9$ und $[6]_9$.

×	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8
2	0	2	4	6	8	1	3	5	7
3	0	3	6	0	3	6	0	3	6
4	0	4	8	3	7	2	6	1	5
5	0	5	1	6	2	7	3	8	4
6	0	6	3	0	6	3	0	6	3
7	0	7	5	3	1	8	6	4	2
8	0	8	7	6	5	4	3	2	1

Um $[2]_9 : [4]_9$ zu bestimmen, lesen wir aus der Tabelle die Inverse $[4]_9^{-1} = 7$ heraus und rechnen $[2]_9 \cdot [4]_9^{-1} = [2]_9 \cdot [7]_9 = [14]_9 = [5]_9$.