

Stefan Sokołowski
Matematyka dyskretna

KOMPUTEROWA ARYTMETYKA LICZB CAŁKOWITYCH TO ARYTMETYKA RESZT

Inst. Informatyki Stosowanej, ANS
Elbląg, 2024/2025

Wykład 4, str. 1

Liczenie reszty na kalkulatorze

Kalkulatory mają „tryb naukowy” a w nim funkcję liczenia reszty:



— reszta z 2020 przez 41 wynosi 11.

Liczenie reszty na kalkulatorze

A jeśli kalkulator nie ma takiej funkcji? Wtedy:

$$\text{reszta} = \text{dzielna} - \text{dzielnik} \cdot \text{iloraz}_{\text{całk.}}$$

11		
7	8	9
4	5	6
1	2	3
0	+	-
=	/	×

$$\text{iloraz}_{\text{rzech.}} = 49.268293$$

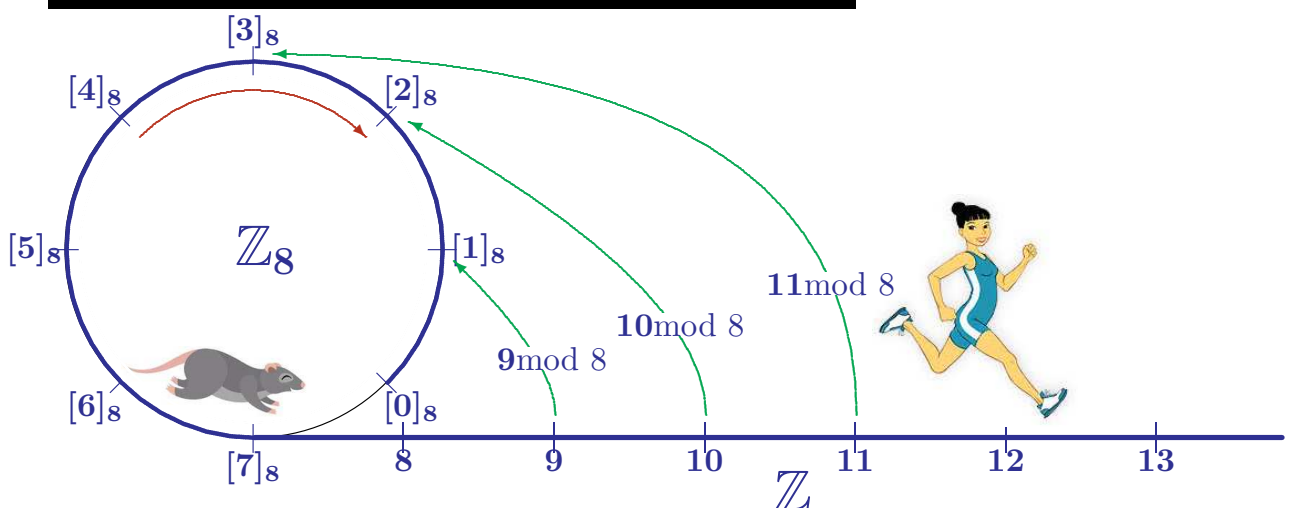
$$\text{iloraz}_{\text{całk.}} = 49$$

w dół!

$$\text{dzielnik} \cdot \text{iloraz}_{\text{całk.}} = 2009$$

$$\text{reszta} = 2020 - 2009 = 11$$

Reszty z dzielenia biegają „w kółko”



$\mathbb{Z}_8$ — klasy liczb o tej samej reszcie:

$$[0]_8 = \{\dots, -16, -8, 0, 8, 16, \dots\}$$

$$[1]_8 = \{\dots, -15, -7, 1, 9, 17, \dots\}$$

$$\dots\dots\dots$$

$$[7]_8 = \{\dots, -9, -1, 7, 15, 23, \dots\}$$

$$[1]_8 = [9]_8 = [-7]_8 = \dots$$

Działania w $\mathbb{Z}_8$

Przekształcenie $\dots \bmod 8 : \mathbb{Z} \rightarrow \mathbb{Z}_8$ wprowadza działania na $\mathbb{Z}_8$:

$$\begin{aligned} [a]_8 + [b]_8 &= [a + b]_8 \\ [a]_8 - [b]_8 &= [a - b]_8 \\ [a]_8 \cdot [b]_8 &= [a \cdot b]_8 \end{aligned}$$

Przykład:

$$[7]_8 + [6]_8 = [13]_8 = [5]_8$$

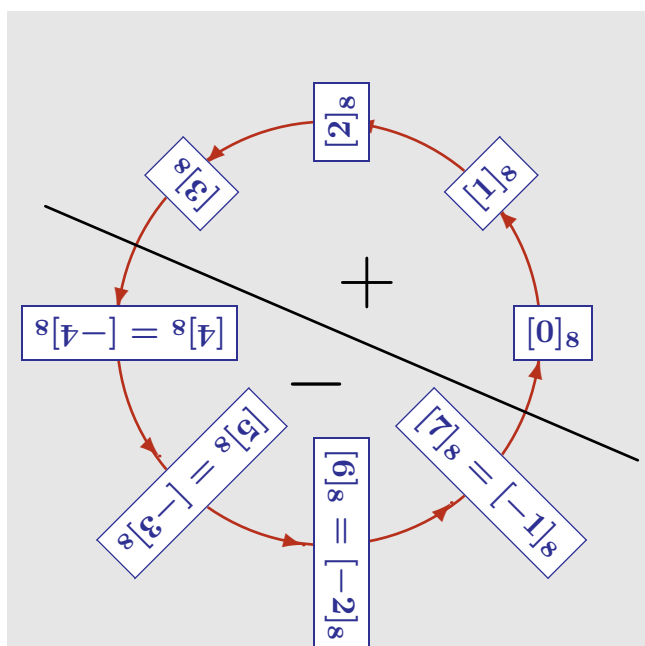
$$[3]_8 - [5]_8 = [-2]_8 = [6]_8$$

$$[5]_8 \cdot [7]_8 = [35]_8 = [3]_8$$

$$\begin{aligned} [35]_8 &= [35]_8 - [4]_8 \cdot [0]_8 \\ &= [35]_8 - [4]_8 \cdot [8]_8 \\ &= [35]_8 - [4 \cdot 8]_8 \\ &= [35 - 32]_8 \\ &= [3]_8 \end{aligned}$$

Komputerowy int

Gdyby na `int` przeznaczać po 3 bity, to typ `int` zawierałby $2^3 = 8$ różnych liczb. Arytmetyka takich liczb: $\mathbb{Z}_{2^3}$.



$[0]_8$	$=$	$\boxed{000}$	
$[1]_8$	$=$	$\boxed{001}$	
$[2]_8$	$=$	$\boxed{010}$	
$[3]_8$	$=$	$\boxed{011}$	
$[-4]_8$	$=$	$\boxed{100}$	} ← ujemne
$[-3]_8$	$=$	$\boxed{101}$	
$[-2]_8$	$=$	$\boxed{110}$	
$[-1]_8$	$=$	$\boxed{111}$	

Komputerowy int

Naprawdę na `int` przeznaczają się (zwykle) 4 bajty czyli 32 bity. Typ `int` zawiera $2^{32} \approx 4 \text{ mld.}$ liczb. Arytmetyka komputerowa: $\mathbb{Z}_{2^{32}}$.

Póki liczymy na małych liczbach — wszystko jak w $\mathbb{Z}$.

W okolicach 2 mld. — należy uważać.

```
#include<stdio.h>

int main() {
    int a = 65536, b = 65535;
    printf(" %d * %d = %d\n", a, b, a*b);
    return 0;
}
```

Ten program mnoży przez siebie dwie liczby dodatnie i drukuje:

$65536 * 65535 = -65536$

(proszę osobiście sprawdzić).

Dlaczego?

$$65536 = 2^{16} \quad 65535 = 2^{16} - 1$$

$$\begin{aligned} [65536]_{2^{32}} \cdot [65535]_{2^{32}} &= [2^{16}]_{2^{32}} \cdot ([2^{16} - 1]_{2^{32}}) \\ &= [2^{16}]_{2^{32}} \cdot [2^{16}]_{2^{32}} - [2^{16}]_{2^{32}} \cdot 1 = [2^{32}]_{2^{32}} - [2^{16}]_{2^{32}} \\ &= [0]_{2^{32}} - [65536]_{2^{32}} = [-65536]_{2^{32}} \end{aligned}$$