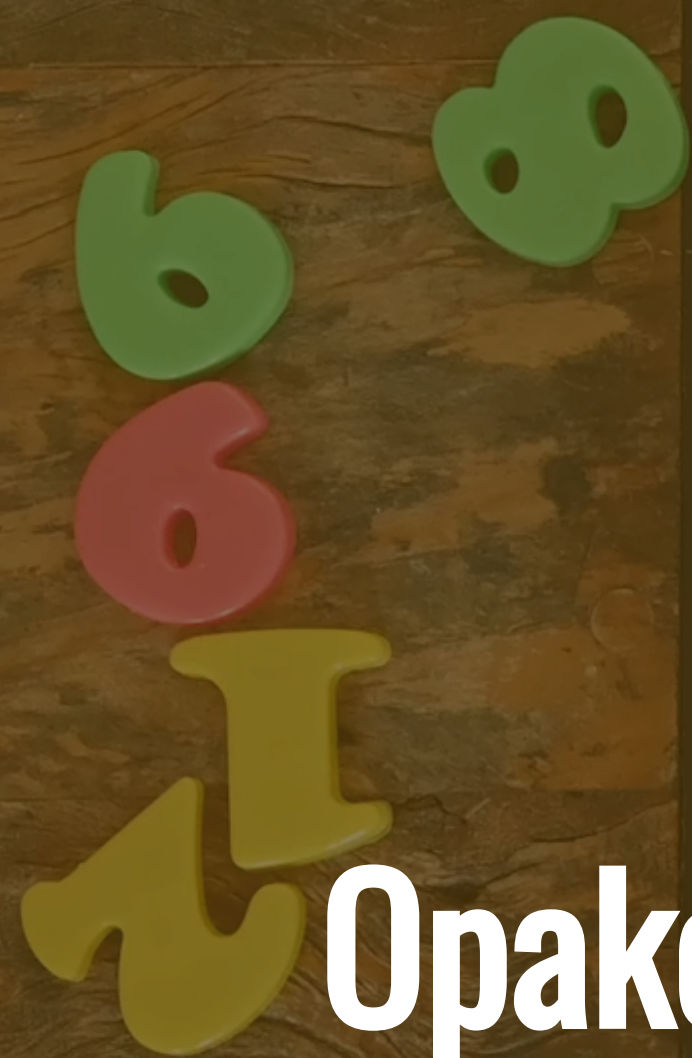


Opakování a asymptotická notace

SSPŠ
OSE4 Pokročilá informatika
2025 / 2026
Ing. Matěj Cajthaml





Opakování matematiky

Aneb jak informatika je matematika

Množina

- Neuspořádaná struktura prvků
- Opakování je nepřipustné
- Může být prázdná

$$\emptyset = \{\} \quad \{1, 3\} \quad \{\square, \triangle\}$$

$$\{\{\square\}, \triangle\} \quad \{1, 1, 2\} = \{1, 2\}$$

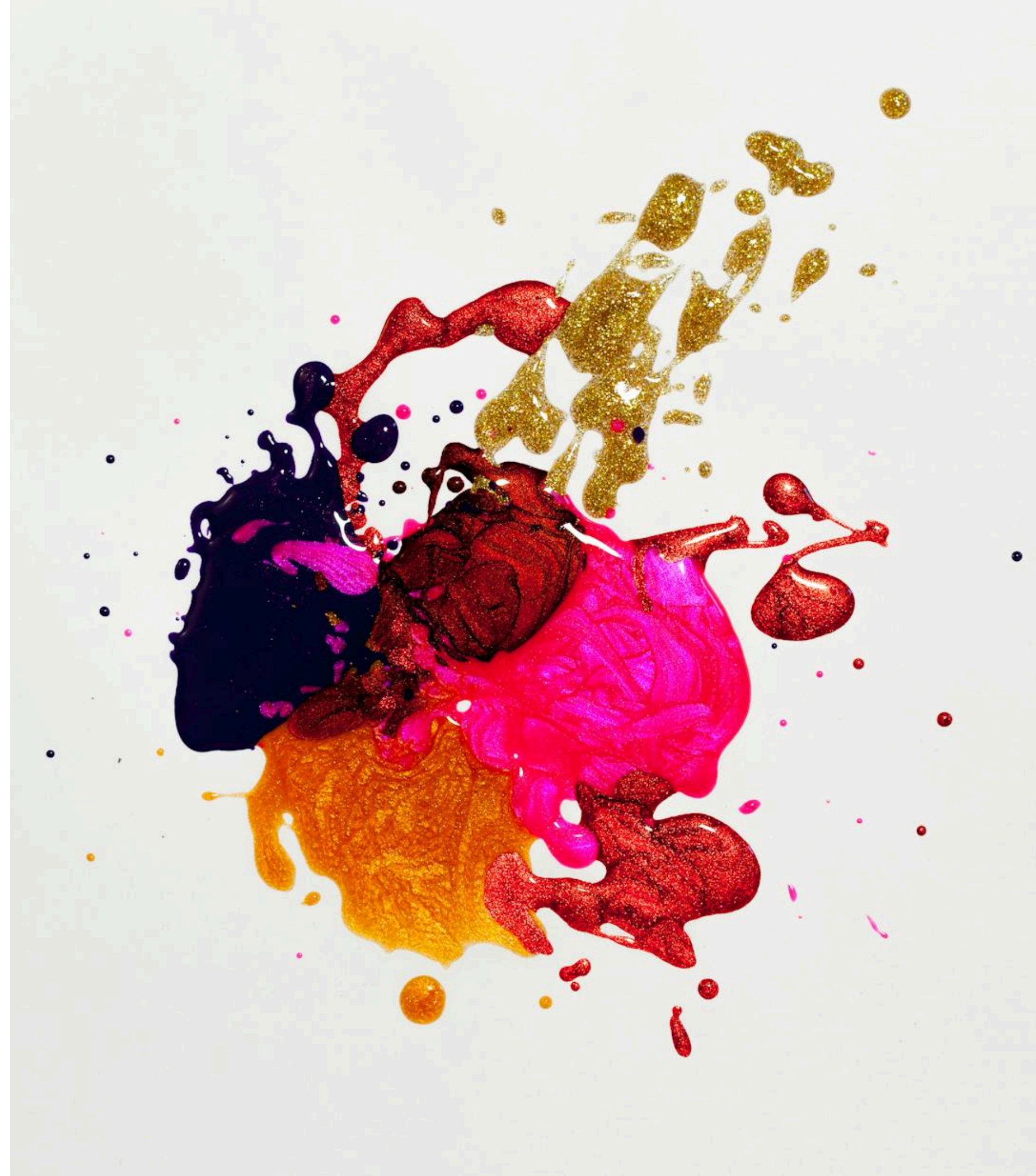
- Číselné obory

$$\mathbb{N}_0 = \{0, 1, 2, 3, \dots\}$$

$$\mathbb{N} = \{1, 2, 3, \dots\}$$

$$\mathbb{Z} = \{\dots, -1, 0, 1, 2, \dots\}$$

$$\mathbb{R} = (-\infty, +\infty)$$



Práce s množinami

$$A = \{1, 3, 4, \square\} \quad B = \{3, \infty, 4, \square\}$$

- Sjednocení

$$B \cup A = A \cup B = \{1, 3, 4, \infty, \square\}$$

- Průnik

$$B \cap A = A \cap B = \{3, 4, \square\}$$

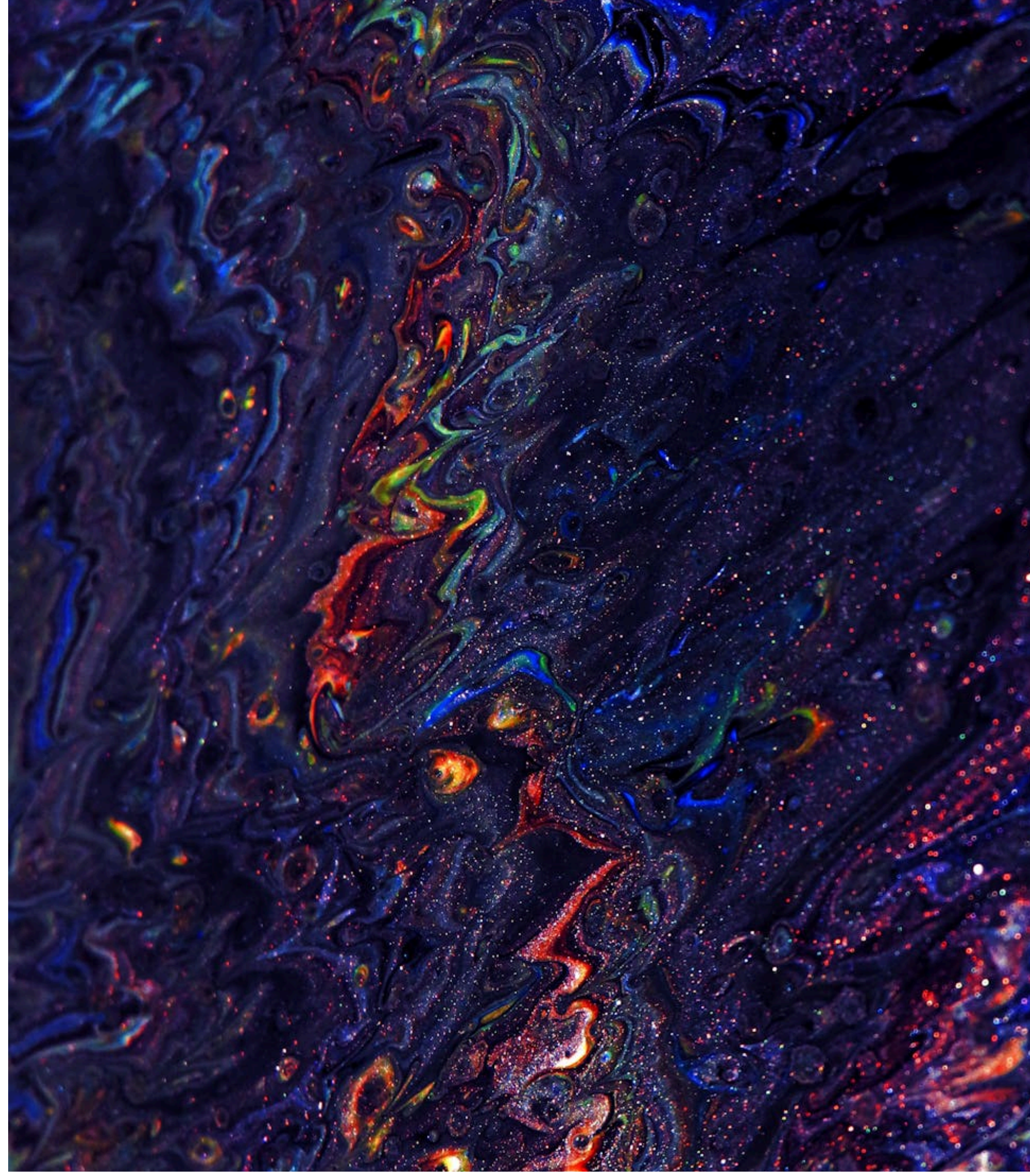
- Rozdíl

$$A \setminus B = \{1\}$$

$$B \setminus A = \{\infty\}$$

- Velikost množiny

$$|A| = 4$$



Funkce

- Vždy definujeme:
- $\Rightarrow$ vstup (definiční obor)
- $\Rightarrow$ výstup (obor hodnot)

$$f : A \rightarrow B$$

- Reálná funkce
- Předpisy

$$f(x) = \begin{cases} 0 & x \leq 0 \\ \frac{100-x}{100} & 0 \leq x \leq 100 \\ 0 & 100 \leq x \end{cases}$$



Logika

- Proměnné A, B, ...
- Spojky $\wedge, \vee, \rightarrow, \leftrightarrow, \neg$

a	b	$(a \wedge b)$
F	F	F
F	T	F
T	F	F
T	T	T

a	b	$(a \rightarrow \neg b)$
F	F	T
F	T	T
T	F	T
T	T	F

a	b	c	$((a \leftrightarrow \neg b) \vee c)$
F	F	F	F
F	F	T	T
F	T	F	T
F	T	T	T
T	F	F	T
T	F	T	T
T	T	F	F
T	T	T	T

Predikátová logika

- Kvantifikátory $\forall, \exists$

$$\exists x \in \mathbb{R} \quad x > 1$$

$$\forall x \in \mathbb{N} \quad \exists y \in \mathbb{N} \quad x \cdot y = x$$





Asymptotická složitost

Aneb jak vidět do nekonečna

OTÁZKA

Co znamená asymptota?

Asymptotická složitost

- Matematický model
- Popisuje chování algoritmu
- Závislost na vstupu
- V nekonečnu - "když se vstup zvětšuje"
- Porovnání algoritmů
- Odhad výpočetního času a paměti
- Nutné konzultovat praxi
- Čas => počet provedených operací
- Paměť => počet použitých buněk



Notace

- Landauova notace

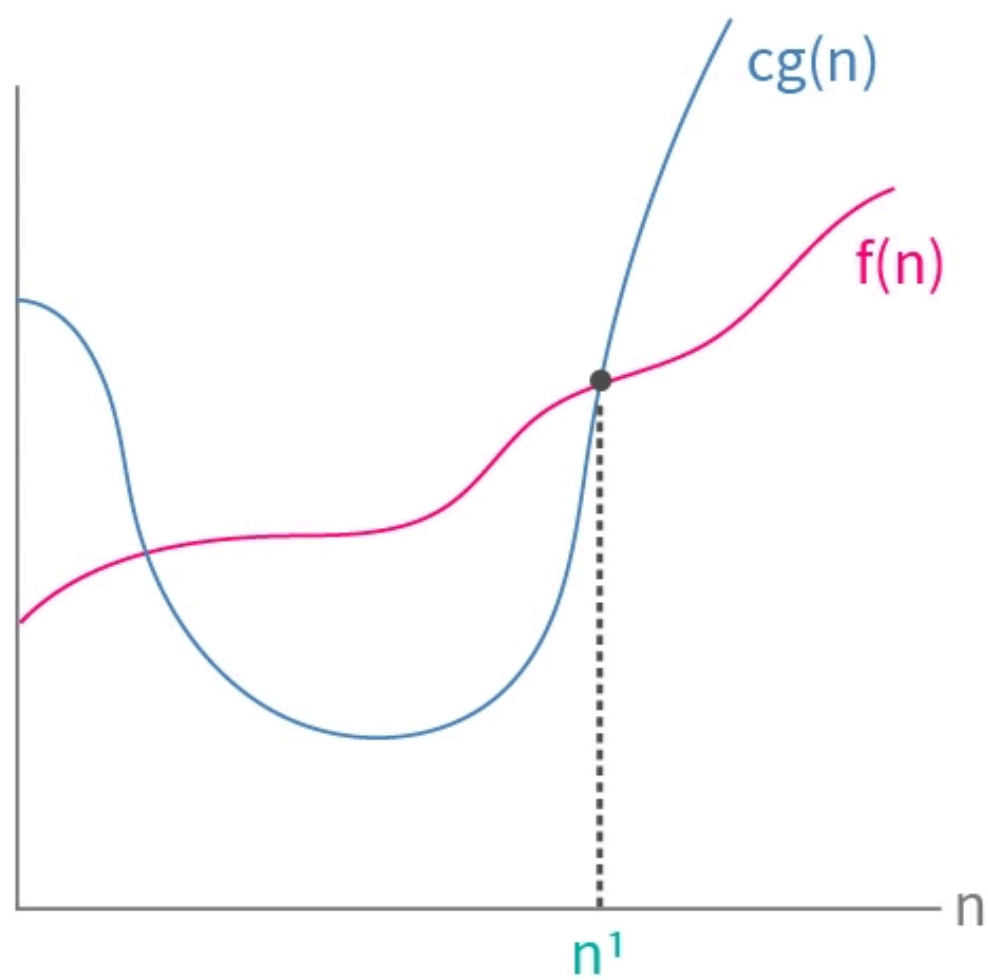
$$O \quad \exists c \in \mathbb{R}^+, \exists n_0 \in \mathbb{N}, \forall n \geq n_0 : f(n) \leq c \cdot g(n)$$

$$o \quad \forall c \in \mathbb{R}^+, \exists n_0 \in \mathbb{N}, \forall n \geq n_0 : f(n) < c \cdot g(n)$$

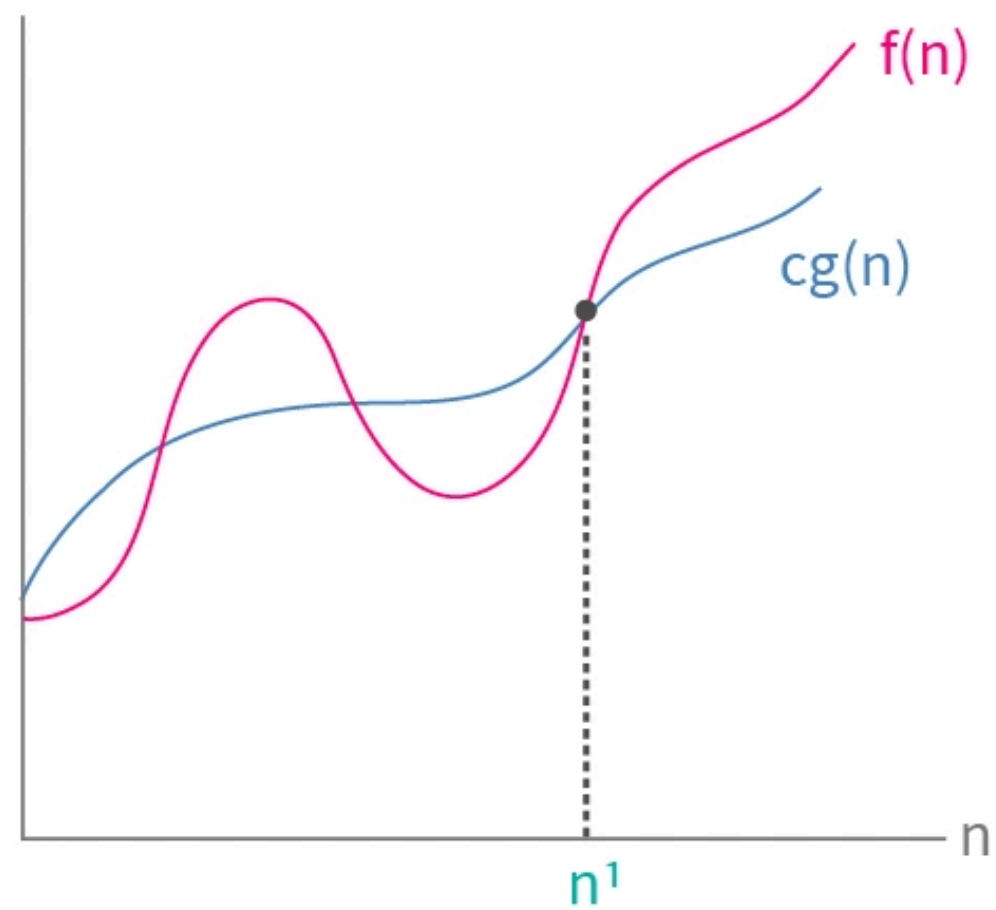
$$\Omega \quad \exists c \in \mathbb{R}^+, \exists n_0 \in \mathbb{N}, \forall n \geq n_0 : f(n) \geq c \cdot g(n)$$

$$\omega \quad \forall c \in \mathbb{R}^+, \exists n_0 \in \mathbb{N}, \forall n \geq n_0 : f(n) > c \cdot g(n)$$

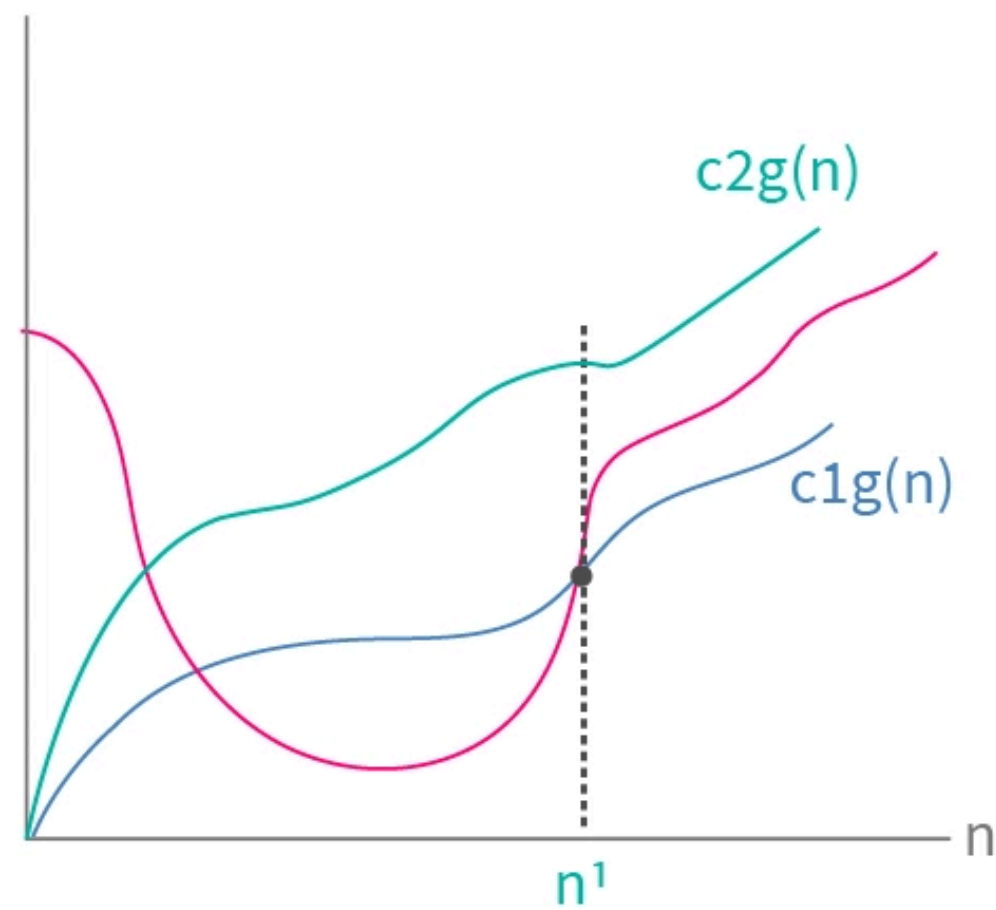
$$\Theta \quad \exists c_1, c_2 \in \mathbb{R}^+, \exists n_0 \in \mathbb{N}, \forall n \geq n_0 : c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$$



$$f(n) = o(g(n))$$



$$f(n) = \Omega(g(n))$$



$$f(n) = \Theta(g(n))$$

Klasické složitosti

$O(1)$ konstantní složitost

$O(\log n)$ logaritmická složitost

$O(n)$ lineární složitost

$O(n \log n)$ lineárně logaritmická složitost

$O(n^2)$ kvadratická složitost

$O(n^3)$ kubická složitost

Chování velkého 0

$$2 \in O(1)$$

$$2 \cdot n^2 + 3 \in O(n^2)$$

$$\frac{x^2 + 3x + 2}{x + 1} \in O(x)$$

$$2^{n+1} \in O(2^n)$$

$$3 + 2 \cdot n \cdot \log n \in O(n \cdot \log n)$$

Chování dalších notací

$$2 \in o(n)$$

$$2 \in o(n!)$$

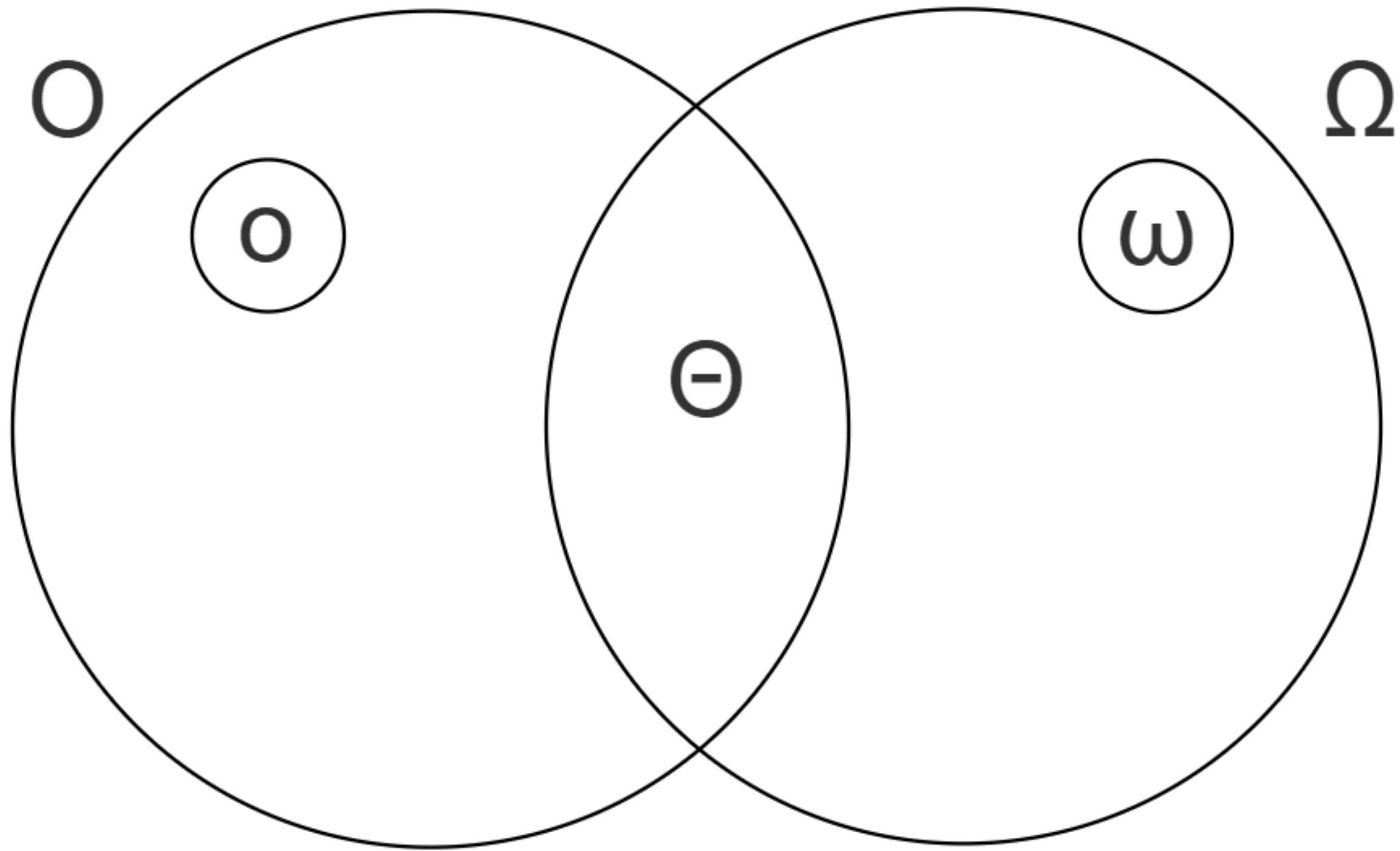
$$n \in \Omega(1)$$

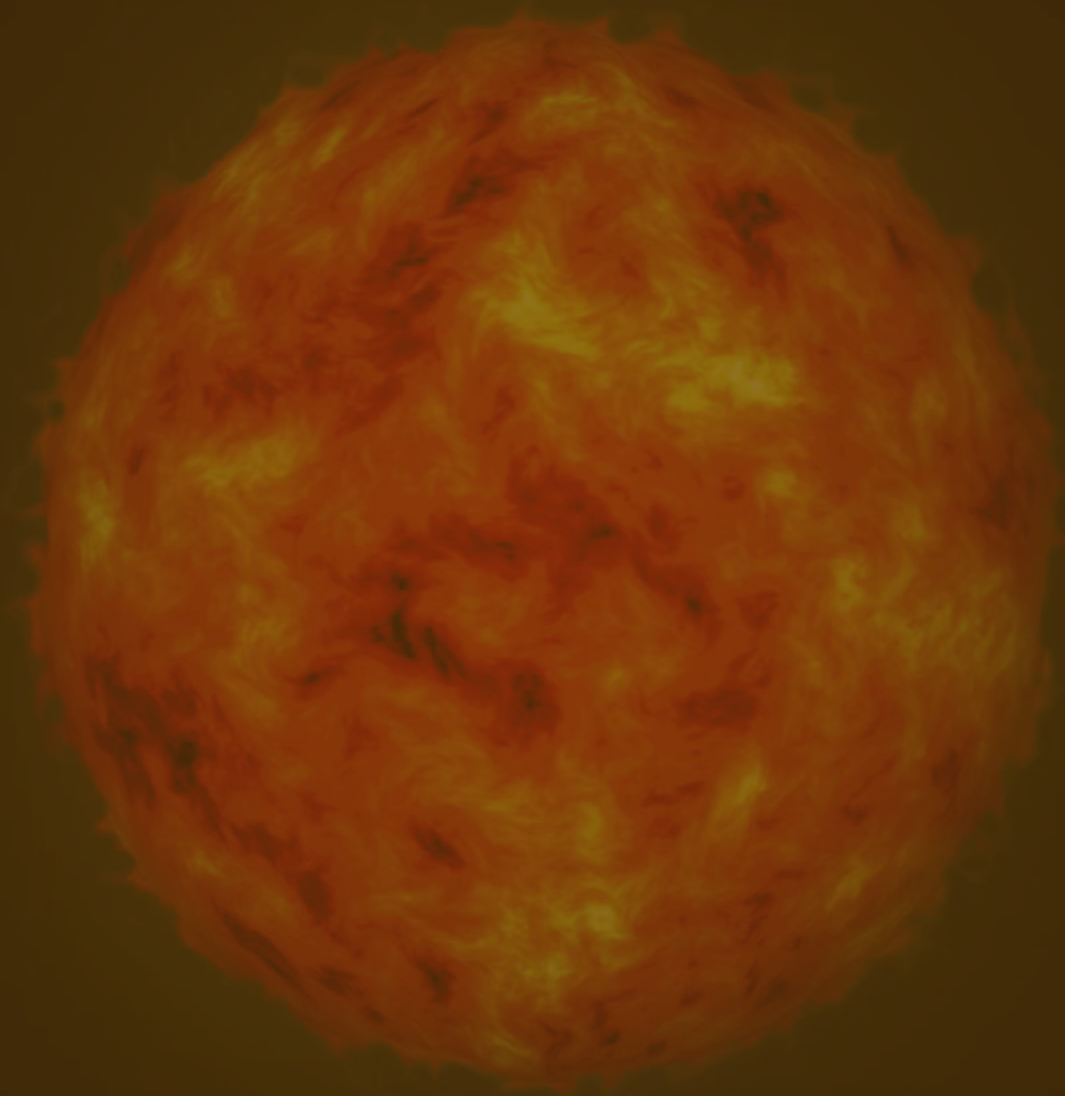
$$n \in \Omega(\log n)$$

$$n \in \omega(1)$$

$$n^2 \in \Theta(n^2)$$

$$5n^3 + 2n^2 + 300 \in \Theta(n^3)$$





Procvičování

Aneb hvězdy i ve snech

- Kolik hvězd?

HVĚZDY 1

```
1 for i:=1 to n do
2     for j:=1 to n do
3         print "*"

```

HVĚZDY 2

```
1 for i:=1 to n do
2     for j:=1 to i do
3         print "*"

```

HVĚZDY 3

```
1 while n > 0 do
2     print "*"
3     n := n/2
```

HVĚZDY 4

```
1 while n > 0 do
2     if (n is odd) then
3         print "*"
4     n := n/2
```


HVĚZDY 5

```
1  rekurze(m):
2      if n = 0 then
3          return 0
4      else
5          return rekurze(m-1) +
6
7  for i:=1 to rekurze(n):
8      print "*"

```

HVĚZDY 6

```
1  rekurze(m):
2      if n <= 0 then
3          return 0
4      else
5          return rekurze(m/2) + 1
6
7  for i:=1 to rekurze(n):
8      print "*"

```

Opačné procvičování

- Vymyslete algoritmy v pseudokódu, které budou mít následující asymptotické složitosti:

$$O(1)$$

$$O(n^3)$$

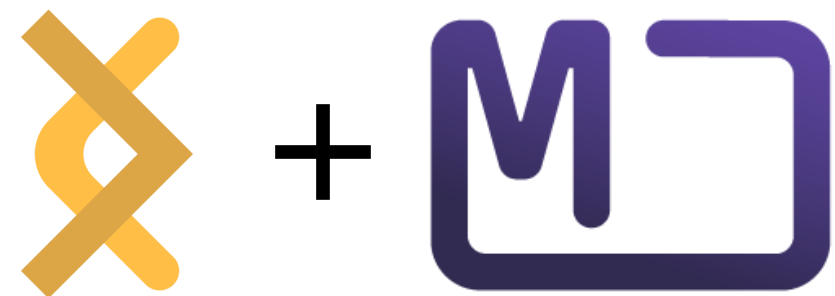
$$O(n \log n)$$

$$O(n!)$$

Děkuji za pozornost

matej.cajthaml@ssps.cz

ssps.cajthaml.eu



Vytvořeno pomocí aplikace Materialist

© 2025 Matěj Cajthaml