

# Posloupnosti a vektory

E 3011

Jan Böhm

RECETOX

April 3, 2024



# Co nás dnes čeká

1 Fibonachos

2 Modely růstu populace

3 Vektorová algebra

4 

# Fibonacciho posloupnost

## Definice

Fibonacciho posloupnost začíná čísly 1, 1 a každý další člen je součtem předchozích dvou, tedy 1, 1, 2, 3, 5, 8, ....

## Fibonacci

Vytvořte funkci `fibonacci(n)`, která vrátí list obsahující prvních  $n$  členů Fibonacciho posloupnosti.

Hint: záporné indexy.

# Co nás dnes čeká

1 Fibonachos

2 Modely růstu populace

3 Vektorová algebra

4 

## Malthusovský růst

Nejjednodušší model růstu populace je geometrická posloupnost.

Naprogramujte funkci `geomSeq(a0, q, n)`, která vrátí list obsahující prvních  $n$  členů geometrické posloupnosti s prvním členem  $a_0$  a kvocientem  $q$ .

Poté modelujte růst celosvětové populace – v roce 2024 žije na zemi cca 8 miliard lidí a každoročně se tato populace zvětší o 1.08 %.

Kdy můžeme očekávat, že velikost celosvětové populace dle tohoto modelu přesáhne 10 miliard?

## Model s kapacitou prostředí

Realističtější model počítá s kapacitou prostředí:

$$x(t+1) = x(t) + r \cdot x(t) \cdot \left(1 - \frac{x(t)}{K}\right),$$

kde  $x(t)$  značí velikost populace v čase  $t$ ,  $K$  je kapacita prostředí a  $r$  koeficient růstu.

Napište funkci `verhulst(x0, r, K, n)`, která vrací list prvních  $n$  členů posloupnosti předpovídaných velikostí populace o počáteční velikosti  $x_0$  a koeficientu růstu  $r$  v prostředí s kapacitou  $K$ .

Model pak spusťte s parametry:  $x_0 = 8$ ,  $K = 12$ ,  $r = 0.08$  a  $n = 100$ . Pomocí kódu na dalším slidu pak výsledek vizualizujte.

Nejprve přes Tools -> Manage Packages nainstalujte matplotlib.  
Poté:

```
1 from matplotlib.pyplot import plot, show
2
3 x = list(range(100))
4 y = verhulst(x0 = 8, r = 0.08, K = 12, n = 100)
5 plot(x, y)
6 show()
```



# Co nás dnes čeká

1 Fibonachos

2 Modely růstu populace


3 Vektorová algebra

4 🍕

Podívejte se na slajdy o operacích s vektory v přednášce k tomuto týdnu. Měli byste být schopni spočítat všechny operace, které tam jsou zdefinované. Není nutné na všechny tvořit funkce, ale doporučuji si naimplementovat:

- Norma vektoru `norm(u)`.
- Násobení vektoru skalárem `SVP(c, u)`.
- Sčítání vektorů `addVectors(u, v)`.
- Skalární součin `dotProduct(u, v)`.
- Vektorový součin `crossProduct(u, v)` (pouze v trojrozměrném prostoru).
- Odchylku vektorů `angle(u, v)`.

# Co nás dnes čeká

- 1 Fibonachos
- 2 Modely růstu populace
- 3 Vektorová algebra
- 4 



Po dnešním cvičení byste měli umět:

- 1 Pracovat s poli. Vytvořit pole, přidávat a odebírat z něj prvky, najít jeho minimum, maximum, součet. Seřadit pole. Zjistit, zda nějaký prvek v něm leží.
- 2 Vytvořit funkci, která na základě nějakého rekurentního vztahu  $x(t+1) = f(x(t), x(t-1), \dots)$  generuje členy této posloupnosti.
- 3 Mít implementovány základní funkce pro vektorové počítání.