

02 - Praktické cvičení

NMIN112

Tomáš Karella Martin Mareš

11.03.2020

Vítejte

Co nás dnes čeká?

- 1 Zvídavé otázky
- 2 Z minule
- 3 Balíčky
- 4 Dokumentace
- 5 Numpy
- 6 Úkoly

Obrazek: Big O Notation

Zvdave otázky - Protipříklad

Ulohy z minulého cvičení

Vzorové kody najdete na naší stránce v záložce "Materiály".
Zadání budou přetvřeno vce podrobně.

Recodex

Pozor na okrajové podmínky!
Pozor na celočíselné a reálné dělení!
Zopakujte si načtení vstupu!

Varianta 1

```
for line in sys.stdin:  
    up_line = line.upper()  
    print(up_line)
```

Varianta 2

```
while True:  
    try :  
        line = input ()  
        print (line .upper ())  
    except EOFError:  
        break
```

Opakovan

Python balcky se typicky anglicky nazyvaj "Packages".
Balcek se vesinou zabvva konkrétnm problémem a poskytuje nástroje pro práci v tomto kontextu. (Numpy - scienti c computing)

Spřava balcku

Balcky instalujeme pomoc Python Package Installer neboli pip.

Overit verzi pip: `python -m pip {version}`

Instalace balcku: `python -m pip install "SomeProject"`

Pro pokročile

V Pythonu lze vytvřret nezavisá prostředí napřklad pomoc virtualenv.

Obrazek: Tip na o ine dokumentaci

[Obrazek: Odkaz na slidy Martina Mareše](#)

Domác úkol

Invarianty pro dnešek

Všečte pomoc knihovny Numpy.

Indexovan od 0 je super.

Tentokrát nečte načten vstupu a každou ulohu pište jako funkci.

Vstupn parametry volte nahodně.

Zadán

Vyrobte matici tvaru $\begin{matrix} R \\ S \end{matrix}$ vyplnenou csmem c.

Example

Funkce `maticec(r,s,c)` vrat matici (`np.array`), pro vstupy `r=5`, `s=4` a `c=5`:

```
[[5. 5. 5. 5.]  
 [5. 5. 5. 5.]  
 [5. 5. 5. 5.]  
 [5. 5. 5. 5.]  
 [5. 5. 5. 5.]
```

Zadan

Vyrobte matici tvaru $r \times s$, která má na okrajích jedničky a jinde nuly.

Example

Funkce `matices_okrajem(r,s)` vrat matici (`np.array`), pro vstupy `r=5, s=4`:

```
[[1.  1.  1.  1.]  
 [1.  0.  0.  1.]  
 [1.  0.  0.  1.]  
 [1.  0.  0.  1.]  
 [1.  1.  1.  1.]]
```

Zadán

Vyrobte matici tvaru $R \times S$, která bude mít v i -tém řádku samé číslo i .

Example

Pro vstupy $r=5$, $s=4$ matice_ocislovanymiradky vrac:

```
[[0. 0. 0. 0.]  
 [1. 1. 1. 1.]  
 [2. 2. 2. 2.]  
 [3. 3. 3. 3.]  
 [4. 4. 4. 4.]]
```

Zadání

Spočítejte determinant 10 matic 10×10 vyplněných náhodnými čísly mezi -1 a 1 (viz np.linalg.det)

Example

Na výstupu očekáváme list dvojice matice a jejího determinantu.
(Ilustrativní příklad má matici velikosti 3×3).

```
( (array ([
      [ 0.28,  0.33,  0.99 ],
      [ 0.91,  0.41,  0.94 ],
      [ 0.71,  0.20,  0.68 ]
]),
 0.0969), ... ]
```


Zadan

Uvažujme "přeměrovac Fibonacciho čísla" de novara rekurentním vztahem $x_i = (x_{i-1} + x_{i-2})=2$. Vme-li, že $x_8 = 426$ a $x_9 = 427$, kolik je x_0 a x_1 ?

Jak na to?

Vyřešte soustavu lineárních rovnic pomocí `np.linalg.solve`. Pro výrobu matic s jedničkami na posunutě diagonále se hod `np.eye`.

Prklad s np.linalg.solve

Prklad pouzt np.linalg.solve prošen rovnic $x_0 + 2x_1 = 1$
a $3x_0 + 5x_1 = 2$:

```
>>> a = np.array([[1, 2], [3, 5]])  
>>> b = np.array([1, 2])  
>>> np.linalg.solve(a, b)  
array([= 1.,  1.]
```

Example

Na vystupu tedy œekavame dvojici $[x_0, x_1]$ "přumerovac Fibonacciho
čísle". Pro $x_8 = 532$ a $x_9 = 531$:

```
[702. 446.]
```

Bonus 1

Vyrobte booleovskou matici, která má na pozici (i,j) True právě tehdy, je-li i dělitelné j .

Bonus 2

Vyrobte vektor, který pro každé číslo od 1 do N uvádí počet jeho dělitelů.

Bonus 3

Vyrobte vektor všech prvočísel mezi 1 a N .

