Posloupnosti a vektory

E 3011

Jan Böhm

RECETOX

April 12, 2023

Co nás dnes čeká

1. Pomocné funkce
2. Hlavní funkce
3. Aplikace
4. 🍕

Pomocné funkce I

Stáhněte si soubor `matrixSupportFunctions.py`. Naleznete v něm kostry kódů pro tvorbu pomocných funkcí.

`zeros(nrow, ncol)`

Vytvořte funkci `zeros(nrow, ncol)`, která vrací matici plnou nul s `nrow` řádky a `ncol` sloupci.

```
1 def zeros(nrow, ncol):
2     R = [[0 for ???] for ???] # write your code instead of ???
3     return R
4
5 print(zeros(3,4))
6
7 >>> [[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]]
```

Pomocné funkce II

size(M)

Vytvořte funkci `size(M)`, která vrací vektor délky 2 s počtem řádků a počtem sloupců matice.

```
1 def size(M):
2     s = [0, 0] # init vector of length 2
3     s[0] = len(???) # write your code instead of ???
4     s[1] = len(???) # write your code instead of ???
5     return s
6
7 print(size(zeros(3,4)))
8
9 >>> [3, 4]
```

Pomocné funkce III

+ × checks

- Vytvořte funkci `canAdd(A, B)`, která vrací `True`, pokud lze matice `A, B` sečíst a jinak `False`
- Vytvořte funkci `canMultiply(A,B)` která vrací `True`, pokud lze matice `A, B` vynásobit (v tomto pořadí) a jinak `False`

```
1  def canAdd(A,B):
2      sizeA = size(A)
3      sizeB = size(B)
4      return ??? and ??? # write your code instead of ???
5
6  def canMultiply(A,B):
7      sizeA = size(A)
8      sizeB = size(B)
9      return ??? # write your code instead of ???
10
11 print(canAdd(A,B), canMultiply(A,B))
12 >>> False True
```

Pomocné funkce IV

výběr řádku nebo sloupce

- Vytvořte funkci `getRow(A, i)`, která vrací `i`-tý řádek matice `A`
- Vytvořte funkci `getCol(A, j)`, která vrací `j`-tý sloupce matice `A`

Indexujeme od 0.

```
1  def getRow(A, i):
2      return ???
3
4  def getCol(A, j):
5      sizeA = size(A)
6      col = []
7      for i in range(???):
8          col.append(???)
9      return col
10
11 print(getRow(A, 0)) >>> [1, 2, 3]
12 print(getCol(A, 1)) >>> [2, -1, 0, 1]
```

Pomocné funkce V

Transponování matice

Vytvořte funkci `transpose(A)`, která vrátí transponovanou matici `A` (vyměněné řádky a sloupce).

```
1  def transpose(A):
2      sizeA = size(A)
3      # R is matrix of zeros in size of transposed A
4      R = zeros(???) # write your code instead of ???
5      # fill entries in R with values in A
6      for i in range(???): # write your code instead of ???
7          for j in range(???): # write your code instead of ???
8              R[i][j] = ??? # write your code instead of ???
9      return R
10
11 print(transpose(A))
12 >>> [[1, 1, 4, 1], [2, -1, 0, 1], [3, 0, 2, 1]]
```

Pomocné funkce VI

vynechání sloupce

Vytvořte funkci `omittCol(A, j)`, která vrátí matici `A` s vynecháním `j`-tým sloupcem.
K čemu by se nám tato pomocná funkce mohla hodit?

```
1  def omittCol(A, j):
2      sizeA = size(A)
3      # init matrix of correct size
4      B = zeros(???, ???) # write your code instead of ???
5      # for each row in A
6      for i in range(???):
7          a = ???
8          del a[j]
9          B[i] = ???
10     return ???
11
12 print(omittCol(A, 1))
13 >>> [[1, 3], [1, 0], [4, 2], [1, 1]]
```

Co nás dnes čeká

1. Pomocné funkce
2. Hlavní funkce
3. Aplikace
4. 🍕

Hlavní funkce

Hlavní funkce

Naimplementujte tyto 3 funkce:

- `addMatrices(A, B)`, která ověří, zda lze matice `A, B` sečíst a pokud ano, vrátí jejich součet.
- `numberTimesMatrix(c, A)`, která vynásobí matici `A` skalárem `c` a tento součin vrátí.
- `matrixProduct(A, B)`, která která ověří, zda lze matice `A, B` v tomto pořadí vynásobit a pokud ano, vrátí jejich součin.
- `matrixVectorProduct(A, u)`, který spočítá `Au` - násobení vektoru `u` maticí A zleva a výsledek vrátí jako vektor.

První dvě funkce jsou jednoduché, pro třetí funkci je návod na dalším slajdu. Čtvrtá funkce pak už bude docela snadná.

Součin matic

Na vstupu jsou dvě matice `A`, `B`.

- Ověřte, že `A` a `B` lze vynásobit. Pokud ne, napište chybu a funkce vrátí `None`. Jinak:
  - Vytvořte si matici 0 správných rozměrů, např. `R`.
  - Pro každý řádek v `A`
    - Pro každý sloupec v `B`

      Spočítejte skalární součin a vložte na správné místo v `R`.
  - Vraťte `R`.

Co nás dnes čeká

1. Pomocné funkce
2. Hlavní funkce
3. Aplikace
4. 🍕

Transformace v $\mathbb{R}^2$

Rotace v $\mathbb{R}^2$

Pomocí násobení vhodnou maticí dokážeme rotovat bod (x, y) v rovině okolo počátku souřadnicového systému. Implementujte funkci `rotate(x, angle)`, která bod `x = [x0, y0]` orotuje okolo počátku o úhel `angle` (ve stupních) v kladném směru otáčení.

```
1 print(rotate([0,1], 90))
2 >>> [-1.0, 6.123233995736766e-17]
3
4 print(rotate([2,3], -30))
5 >>> [3.23205080756887, 1.598076211353316]
```

Co nás dnes čeká

1. Pomocné funkce
2. Hlavní funkce
3. Aplikace
4. 🍕

Takeaways

Po dnešním cvičení byste měli umět:

- Pracovat s dvourozměrnými poli. Rozumět indexování dvourozměrných polí, umět vybrat nějakou podmnožinu hodnot z takového pole.

Po dnešním cvičení byste měli mít připravené tyto funkce:

- Sadu pomocných funkcí pro manipulaci s maticemi.
- Funkci pro sčítání matic, násobení matic a násobení vektoru maticí.