

Wtorek, 27 marca 2018

Ćwiczenie 04

1. Proszę napisać program testujący alternatywne sposoby budowania list wyników: pętla for, lista składana, funkcja *map* i wyrażenie generatora (składnia taka jak listy składanej tylko w miejsce nawiasów kwadratowych należy wstawić okrągłe). Dla każdego ze sposobów proszę utworzyć osobną funkcję tak, aby uzupełnić poniższy kod: `from time import time`
`from sys import version`

```
powt=1000
```

```
N=10000
```

```
(...)
```

```
print(version)
```

```
test=(forStatement, listComprehension, mapFunction, generatorExpression)
```

```
for testFunction in test:
```

```
    print(testFunction.__name__.ljust(20), '=>', tester(testFunction))
```

gdzie: *tester* - funkcja wywołująca *powt* razy daną funkcję, w której tworzona jest lista o rozmiarze *N*. W ramach testu najpierw proszę tylko dodawać elementy a następnie przy ich dodawaniu proszę je podnieść do kwadratu. Do pomiaru czasu proszę użyć funkcji *time* z modułu *time* (`help(time)`). Proszę sprawdzić czy wynik zależy od wersji Pythona! (2p)

2. Proszę napisać funkcję obliczającą wartość całki metodą trapezów korzystając z funkcji *map* (funkcja podcałkowa, granice całkowania oraz liczba kroków jako parametry wywołania funkcji) (2p)

3. Proszę wyznaczyć wartość liczby *pi* metodą Monte-Carlo korzystając z funkcji *filter* (2p).

4. Proszę znaleźć:

- największą wartość w każdym wierszu macierzy (*map*),
- największą wartość w każdej kolumnie macierzy (*map+zip*),
- sumę dwóch macierzy.

5. Każde polecenie jedna linijka (2p)

6. Proszę napisać funkcję przyjmującą dwa parametry - lista x-ów i y-ów. Korzystając z funkcji *reduce* i *map* proszę obliczyć (i zwrócić z funkcji) wartości dofitowanych współczynników prostej oraz ich niepewności (2p).

$$a=(1/D)\sum y_i(x_i-x);$$

$$D=\sum (x_i-x)^2;$$

$$b=y-ax$$

$$\Delta y=\sqrt{(\sum (y_i-(ax_i+b))^2)/(n-2)};$$

$$\Delta a=\Delta y/\sqrt{D};$$

$$\Delta b=\Delta y \sqrt{1/n+x^2/D}$$