

Laboratorium 13 — Opracowanie programów realizujących obliczenia inżynierskie z wykorzystaniem biblioteki NumPy i SciPy.

NumPy jest główną biblioteką stosowaną do obliczeń inżynierskich w Pythonie. Dostarcza wydajne wielowymiarowe tablice i narzędzia do pracy z tymi tablicami. Aby pracować z biblioteką należy ją zaimportować i ewentualnie przypisać jej alias:

```
import numpy as np
```

Tablice

Główną klasą tablic w NumPy jest `ndarray` (alias `array`), która w konstruktorze przyjmuje jeden argument — listę. Przykładowo, aby utworzyć tablicę jednowymiarową, należy:

```
A = np.array([1,2,3]),
```

a dwuwymiarową:

```
A = np.array([[3,4,5,6],[9,16,25,36]])
```

Analogicznie tworzone są tablice o większej liczbie wymiarów. Podczas tworzenia tablicy można także wskazać typ danych jakie ma przechowywać:

```
A = np.array([(1,2), (3,4)], dtype=complex)
```

Główne atrybuty klasy: `ndim`, `shape`, `size`, `dtype`, `itemsize`.

Do tworzenia tablic o specyficznej charakterystyce służą np.:

- `ones` — tablica jedynek,
- `zeros` — tablica zer,
- `empty` — tablica z przypadkowymi wartościami, zależnymi od stanu pamięci,
- `full` — tablica ze stałymi,
- `eye` — tablica z macierzą jednostkową,
- `random.random` — tablica z wartościami losowymi.

Indeksowanie tablic

Tablice mogą być indeksowane na kilka sposobów (indeksowanie od 0):

- `tablica[indeks]` — pojedynczy element,
- `tablica[tablica_indeksow]` — elementy o podanych indeksach,
- `tablica[początek:koniec]` — elementy o indeksach od `początek` do `koniec-1`,
- `tablica[początek:koniec:krok]` — elementy o indeksach od `początek` do `koniec-1` z krokiem `krok`,
- `tablica[tablica_logiczna]` — indeksowanie tablicą logiczną.

Także w przypadku tablic wielowymiarowych, używa się powyższych indeksacji dodając do nich kolejne wymiary po przecinku.

Dodatkowo, aby wypisać np. całą drugą kolumnę pewnej tablicy `A` można zastosować operator `:`
`A[:,1]`

Iterowanie po tablicach

Dwa równoważne sposoby iteracji po tablicy dwuwymiarowej:

```
for r in range(0, len(A)):
    for c in range(0, len(A[0])):
        print(A[r, c])

for row in A:
    for elem in row:
        print(elem)
```

Zadanie 1 Napisz funkcję przyjmującą jeden parametr n — rozmiar macierzy kwadratowej, a zwracającą macierz o zadanej poniżej strukturze:

$$\begin{bmatrix} 1 & 1 & 1^2 & \dots & 1^{n-1} \\ 1 & 2 & 2^2 & \dots & 2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & n & n^2 & \dots & n^{n-1} \end{bmatrix}$$

Zadanie 2 Napisz funkcję przyjmującą jeden parametr n — rozmiar macierzy kwadratowej, wypełnioną w taki sposób, aby kolejne liczby od 1 do n^2 układały się od lewej do prawej w pierwszym wierszu, od prawej do lewej w drugim itd. Przykład dla $n = 4$ został pokazany poniżej.

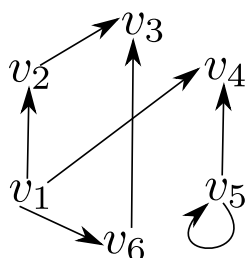
$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 8 & 7 & 6 & 5 \\ 9 & 10 & 11 & 12 \\ 16 & 15 & 14 & 13 \end{bmatrix}$$

Zadanie 3 Napisz funkcję przyjmującą jeden parametr n — rozmiar macierzy kwadratowej, wypełnioną w taki sposób, aby kolejne liczby od 1 do n^2 układały się po spirali. Przykład dla $n = 4$ został pokazany poniżej.

$$\begin{bmatrix} 4 & 5 & 6 & 7 \\ 3 & 14 & 15 & 8 \\ 2 & 13 & 16 & 9 \\ 1 & 12 & 11 & 10 \end{bmatrix}$$

Zadanie 4 Napisz program, który w macierzy tworzonej przez funkcję z Zadania 2 zamieni wszystkie elementy o wartościach podzielnych przez 5 lub 3 na 0 (użyj indeksowania tablicą logiczną).

Zadanie 5 Napisz funkcję, która na podstawie dowolnej listy sąsiedztwa (słownik list) zwróci odpowiadającą jej macierz sąsiedztwa oraz macierz incydencji dla grafu skierowanego. Np. dla poniższego grafu istnieje następująca lista sąsiedztwa:



```
1: 2, 4, 6
2: 3
3:
4:
5: 4, 5
6: 3
```

Macierz przyległości/sąsiedztwa grafu nazywa się macierz S o wymiarach $n \times n$ (n to liczba wierzchołków), gdzie:

$$s_{ij} = \begin{cases} 1 & \text{jeżeli istnieje łuk } (v_i, v_j), \\ 0 & \text{jeżeli nie istnieje łuk } (v_i, v_j). \end{cases}$$

Tak więc, dla powyższego przykładu, macierz sąsiedztwa wygląda w następujący sposób:

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Macierz incydencji grafu nazywa się macierz C o wymiarach $n \times m$ (n to liczba wierzchołków, a m to liczba krawędzi), gdzie:

$$c_{ij} = \begin{cases} 1 & \text{jeżeli wierzchołek } v_i \text{ jest początkiem łuku } j, \\ -1 & \text{jeżeli wierzchołek } v_i \text{ jest końcem łuku } j, \\ 0 & \text{jeżeli wierzchołek } v_i \text{ nie jest incydentny z łukiem } j. \end{cases}$$

(wartość 2, gdy występuje pętla)

Tak więc, dla powyższego przykładu, macierz incydencji wygląda w następujący sposób:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Zadanie 6 Napisz funkcję, która przyjmie jeden argument — dowolną macierz kwadratową A , a zwróci jej wyznacznik. Wyznacznik ma zostać wyznaczony metodą Laplace'a.

<https://www.matmana6.pl/obliczanie-wartosci-wyznacznika-metoda-laplacea#>