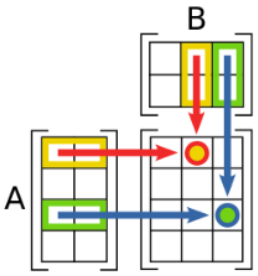
Mnożenie macierzy

liczba kolumn pierwszej macierzy = liczba wierszy drugiej macierzy

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \cdot \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1k} \\ c_{21} & c_{22} & \dots & c_{2k} \\ \dots & \dots & \dots & \dots \\ c_{n1} & c_{n2} & \dots & c_{nk} \end{bmatrix}$$

liczba wierszy wynosi n

liczba kolumn wynosi n



Przykładowo, element c_{12}

$$c_{12} = a_{11} \cdot b_{12} + a_{12} \cdot b_{22}.$$

$$c_{ij} = a_i \cdot b_j^\top,$$

gdzie $b_j^\top$ oznacza transpozycję macierzy b.

Mnożenie macierzy można wykonać na kilka sposobów za pomocą funkcji zawartych w bibliotekach.

```
import numpy as np

A = np.array([2, 1, 1, 1, 3, 6, 4, 5, 5]).reshape(3,3)
B = np.array([1, 0, 5, 2, 1, 6, 0, 3, 0]).reshape(3,3)

C = np.dot(A,B)
print("Mnożenie za pomocą np.dot() \n" ,C)
#Dokumentacja NumPy mówi jednak, że używać np.dot() należy tylko do obliczania
#iloczynu skalarnego dwóch jednowymiarowych wektorów, a nie do mnożenia macierzy.

D = np.matmul(A,B)
E = A@B

print("Mnożenie za pomocą np.matmul() \n", D)
print("Mnożenie za pomocą operatora @ \n", E)
```

```
Mnożenie za pomocą np.dot()
 [[ 4  4 16]
 [ 7 21 23]
 [14 20 50]]
Mnożenie za pomocą np.matmul()
 [[ 4  4 16]
 [ 7 21 23]
 [14 20 50]]
Mnożenie za pomocą operatora @
 [[ 4  4 16]
 [ 7 21 23]
 [14 20 50]]
```

Rząd macierzy

Teoria

Macierz *A* jest rzędu *r*, jeżeli istnieje chociaż jeden różny od zera wyznacznik stopnia *r* utworzony z elementów tej macierzy (przy czym elementy bierze się w tej kolejności, w jakiej są one rozmieszczone w danej macierzy), a wszystkie wyznaczniki tej macierzy stopnia wyższego niż *r* mają wartość zero.
Rząd macierzy oznaczany jest jako *r, R* oraz *rz*.

Chcąc obliczyć rząd macierzy musimy znaleźć największą macierz, której wyznacznik jest różny od zera, wielkość tej niezerowej macierzy będzie szukaną wartością, czyli jeśli największą macierzą, której wyznacznik jest różny od zera jest macierz 3x3 to rząd macierzy jest równy 3.

linalg.matrix_rank()

Zwraca rząd macierzy wykorzystując metodę SVD

Rozkład według wartości osobliwych (algorytm SVD (SVD – z ang. Singular Value Decomposition)) – pewien rozkład macierzy (dekompozycja) na iloczyn trzech specyficznych macierzy.

Jest to metoda matematyczna stosowana m.in. w analizie statystycznej służąca do redukcji wymiaru macierzy.

```
import numpy as np

A = np.array([[6, 1, 1, 4],
              [4, -2, 5, 1],
              [2, 8, 7, 4],
              [2, 2, 1, 3]])

# Rank of a matrix
print("Rank of A:", np.linalg.matrix_rank(A)) #rząd
```

```
Rank of A: 4
```

Wyznacznik macierzy

Wyznacznik macierzy to funkcja określona na macierzach kwadratowych związana z mnożeniem i dodawaniem odpowiednich elementów danej macierzy tak, by otrzymać pojedynczą liczbę. Inaczej mówiąc wyznacznik macierzy jest to liczba rzeczywista przyporządkowana macierzy kwadratowej. Wyznacznik oznaczamy symbolicznie det*A* lub |*A*|.

Obliczanie wyznacznika

Dla macierzy stopnia n=2 wyznacznik liczymy w następujący sposób:

$$\det \begin{bmatrix} a & b \\ c & d \end{bmatrix} = ad - cb$$

Dla macierzy stopnia n=3 wyznacznik liczymy stosując tzw. metodę Sarrusa:

$$\det \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} = \begin{vmatrix} a & b & c & a & b \\ d & e & f & d & e \\ g & h & i & g & h \end{vmatrix} = aei + bfg + cdh - ceg - afh - bdi.$$

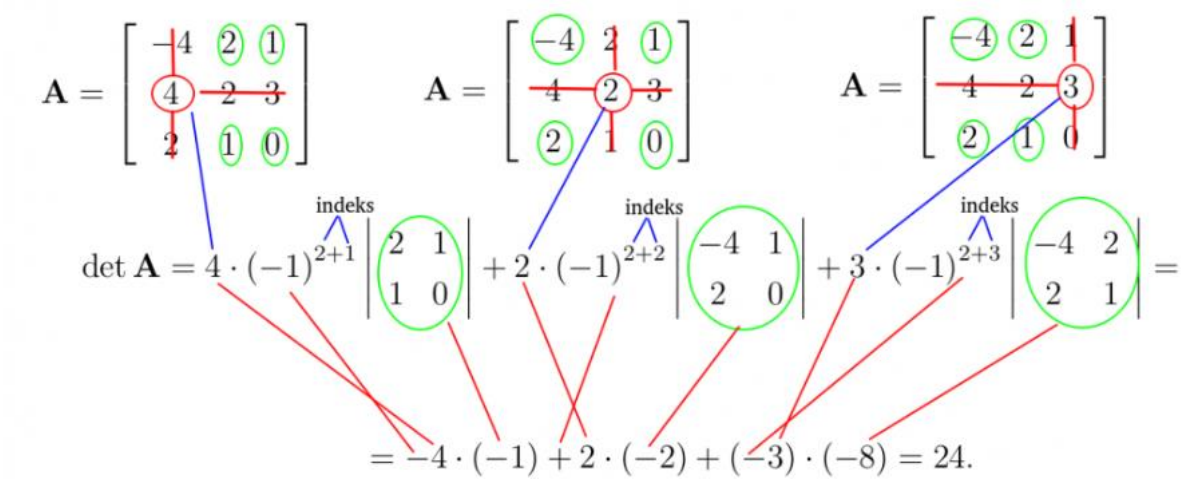
Metoda obliczania wyznacznika wykorzystująca dopełnienie algebraiczne - rozwinięcie Laplace'a

Dopełnienie algebraiczne:

Niech $\mathbf{A} = [a_{ij}]$ *będzie macierzą kwadratową stopnia* $n \geq 2$. *Wówczas* ***dopełnieniem algebraicznym elementu*** a_{ij} *macierzy* $\mathbf{A}$ *nazywamy liczbę*

$$\mathbf{D}_{ij} = (-1)^{i+j} \det \mathbf{A}_{ij},$$

Poniżej obliczam wyznacznik macierzy (det A) rozwinięciem Laplace'a względem 2– ego wiersza:



Funkcje NumPy**.linalg** (skrót od linear algebra) powstały, aby zapewnić wydajne niskopoziomowe implementacje standardowych algorytmów algebry liniowej.

Przykładowa implementacja obliczania wyznacznika przy pomocy moduły **linalg**

```
[16] import numpy as np

     A = np.array([[6, 1, 1],
                   [4, -2, 5],
                   [2, 8, 7]])

     # Determinant of a matrix
     print("\nDeterminant of A:", np.linalg.det(A))
```

```
Determinant of A: -306.0
```

Transponowanie macierzy

Macierz transponowana (**przestawiona**) macierzy *A* – macierz A^T która powstaje z danej macierzy (*w ogólności prostokątnej, w szczególności* jednowierszowej czy o jednej kolumnie) poprzez zamianę jej wierszy na kolumny i kolumn na wiersze. Operację tworzenia macierzy transponowanej nazywa się **transpozycją** (**przestawianiem**).

(1) Transponować można macierz w ogólności prostokątną, np. gdy

$$A = \begin{bmatrix} 2 & 3 & 1 & 4 \\ -1 & 2 & 0 & 1 \\ 2 & 2 & 0 & 1 \end{bmatrix}$$

to macierz transponowana ma postać:

$$A^{\mathrm{T}} = \begin{bmatrix} 2 & -1 & 2 \\ 3 & 2 & 2 \\ 1 & 0 & 0 \\ 4 & 1 & 1 \end{bmatrix}.$$

Istnieje funkcja wbudowana w bibliotece NumPy: **transpose()**

```
import numpy as np

A = np.array([[4, 3, 2, 0],
              [-4, -2, -5, 1],
              [12, 8, -7, 4]])

M = np.transpose(A)
print(M)
```

```
[[ 4 -4 12]
 [ 3 -2  8]
 [ 2 -5 -7]
 [ 0  1  4]]
```

Zadania do wykonania

**Proszę o opisanie każdej linijki kodu

Zadanie 1

Stwórz macierze i wykonaj poniższe działanie:

$$\begin{bmatrix} 2 & 1 & 1 \\ 1 & 3 & 6 \\ 4 & 5 & 5 \end{bmatrix} \text{ x } \begin{bmatrix} 1 & 0 & 5 \\ 2 & 1 & 6 \\ 0 & 3 & 0 \end{bmatrix} = \text{ ?}$$

(To jest tylko przykład, kod ma działać dla każdej macierzy)

Twoja funkcja powinna wykonywać następujące czynności:

- Zaakceptuj dwie macierze, A i B, jako dane wejściowe.
- Sprawdź, czy mnożenie macierzy między A i B jest prawidłowe.
- Jeśli jest poprawna, pomnóż dwie macierze A i B i zwróć macierz iloczynów C.
- W przeciwnym razie zwróć komunikat o błędzie, że macierze A i B nie mogą być pomnożone.

Zadanie 2

Oblicz wyznacznik poniższej macierzy 3x3

$$\begin{bmatrix} 1 & 4 & 5 \\ 2 & 1 & 6 \\ 0 & 3 & 2 \end{bmatrix}$$

(Napisz funkcję, która będzie działała dla każdej wprowadzonej macierzy 3x3)

Zadanie 3

Mając poniższą macierz, napisz funkcję, która będzie odpowiedzialna za jej transponowanie.

$$\begin{bmatrix} 3 & 2 & 4 & 1 & 8 & 0 \\ 2 & 3 & 5 & 6 & 0 & 3 \\ 7 & 7 & 2 & 1 & 4 & 5 \end{bmatrix}$$