

# Zadanie – funkcja obliczająca wyznacznik macierzy

Wyznacznik macierzy ma sens dla macierzy kwadratowych — i jest zdefiniowany rekurencyjnie:

$$\det[a] = a$$

$$\det \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \dots & \dots & & \dots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{bmatrix} =$$

$$= a_{1,1} \det A_{1,1} - a_{1,2} \det A_{1,2} + a_{1,3} \det A_{1,3} - \dots \pm a_{1,n} \det A_{1,n},$$

Gdzie  $A_{i,j}$  to macierz  $A$  z wykreślonym  $i$ -tym wierszem i  $j$ -tą kolumną.

# Zadanie – funkcja obliczająca wyznacznik macierzy

Przykład:

$$\begin{vmatrix} 1 & 0 & 2 & 0 \\ 2 & 3 & 0 & -1 \\ 3 & -1 & -1 & 0 \\ 0 & 1 & -1 & -2 \end{vmatrix} =$$
$$= 1 \begin{vmatrix} 3 & 0 & -1 \\ -1 & -1 & 0 \\ 1 & -1 & -2 \end{vmatrix} - 0 \begin{vmatrix} 2 & 0 & -1 \\ 3 & -1 & 0 \\ 0 & -1 & -2 \end{vmatrix} + 2 \begin{vmatrix} 2 & 3 & -1 \\ 3 & -1 & 0 \\ 0 & 1 & -2 \end{vmatrix} - 0 \begin{vmatrix} 2 & 3 & 0 \\ 3 & -1 & -1 \\ 0 & 1 & -1 \end{vmatrix} =$$

Uwaga ze zmiennością znaków! Mamy na przemian + i – przy kolejnych elementach rozwinięcia (np. powyżej +1, -0, +2, -0)

Podpowiedź do implementacji:  $(-1)^{\text{parzystej}} = 1$ ,  $(-1)^{\text{nieparzystej}} = -1$

# Zadanie – funkcja obliczająca wyznacznik macierzy

Przykład:

$$\begin{vmatrix} 3 & 0 & -1 \\ -1 & -1 & 0 \\ 1 & -1 & -2 \end{vmatrix} = 3 \begin{vmatrix} -1 & 0 \\ -1 & -2 \end{vmatrix} - 0 \begin{vmatrix} -1 & 0 \\ 1 & -2 \end{vmatrix} + (-1) \begin{vmatrix} -1 & -1 \\ 1 & -1 \end{vmatrix}$$

$$\begin{vmatrix} -1 & 0 \\ -1 & -2 \end{vmatrix} = +(-1)|-2| - 0|-1| = 2$$

# Zadanie – funkcja obliczająca wyznacznik macierzy

Jak będziemy zapisywać macierz w kodzie?

$$\begin{pmatrix} 3 & 0 & -1 \\ -1 & -1 & 0 \\ 1 & -1 & -2 \end{pmatrix} \equiv [[3, 0, -1], [-1, -1, 0], [1, -1, -2]]$$

# Zadanie – funkcja obliczająca wyznacznik macierzy

Do wykonania zadania będziemy potrzebowali trzech funkcji:

- a) `def is_matrix_square(mat: List[List[int]]) -> bool:`  
*"""Check if given matrix is a square one."""*
- b) `def get_minor_matrix(mat: List[List[int]], i: int, j: int) -> List[List[int]]:`  
*"""Generate a minor matrix of M for row 'i' and column 'j'."""*
- c) `def matrix_determinant(mat: List[List[int]]) -> int:`  
*"""Compute matrix determinant using Laplace method."""*

# Zadanie – funkcja obliczająca wyznacznik macierzy

Używając podejścia TDD zaimplementuj:

a) `def is_matrix_square(mat: List[List[int]]) -> bool:`  
    *"""Check if given matrix is a square one."""*

Przykłady wywołania:

`is_matrix_square([[2, 4, 2], [3, 1, 1], [1, 2, 0]]) -> True`

`is_matrix_square([[5]]) -> True`

`is_matrix_square([[2, 4], [1, 8], [2, 5]]) -> False`

# Zadanie – funkcja obliczająca wyznacznik macierzy

Używając podejścia TDD zaimplementuj:

```
@pytest.mark.parametrize(
    "input_matrix, expected_result",
    [
        ([[2, 4, 2], [3, 1, 1], [1, 2, 0]], True),
        ([[5]], True),
        ([[2, 4, -3], [1, 8, 7], [2, 5]], False)
    ]
)
def test_is_matrix_square(input_matrix, expected_result):
    assert expected_result == is_matrix_square(input_matrix)
```

# Zadanie – funkcja obliczająca wyznacznik macierzy

Używając podejścia TDD zaimplementuj:

b) `def get_minor_matrix(mat: List[List[int]], i: int, j: int) -> List[List[int]]:`  
    *"""Generate a minor matrix of M for row 'i' and column 'j'."""*

Przykłady wywołania:

```
get_minor_matrix([[3, 0, -1], [-1, -1, 0], [1, -1, 2]], 0, 0) ->  
[[-1, 0], [-1, 2]]
```

```
get_minor_matrix([[3, 0, -1], [-1, -1, 0], [1, -1, 2]], 1, 1) ->  
([[3, -1], [1, 2]])
```



# Zadanie – funkcja obliczająca wyznacznik macierzy

Używając podejścia TDD zaimplementuj:

```
@pytest.mark.parametrize(
    "input_matrix, row, col, expected_result",
    [
        ([[3, 0, -1], [-1, -1, 0], [1, -1, 2]], 0, 0, [[-1, 0], [-1, 2]]),
        ([[3, 0, -1], [-1, -1, 0], [1, -1, 2]], 1, 1, [[3, -1], [1, 2]]),
        (
            [[2, 5, 3, 6, 3],
             [17, 5, 7, 4, 2],
             [7, 8, 5, 3, 2],
             [9, 4, -6, 8, 3],
             [2, -5, 7, 4, 2]],
            2, 3,
            [[2, 5, 3, 3],
             [17, 5, 7, 2],
             [9, 4, -6, 3],
             [2, -5, 7, 2]],
        ),
    ],
)
def test_get_minor_matrix(input_matrix, row, col, expected_result):
    assert expected_result == get_minor_matrix(input_matrix, row, col)
```

# Zadanie – funkcja obliczająca wyznacznik macierzy

Używając podejścia TDD zaimplementuj:

```
c) def matrix_determinant(mat: List[List[int]]) -> int:  
    """Compute matrix determinant using Laplace method. """
```

Flow:

- (1) Sprawdź poprawność danych wejściowych, tj. jeśli mat nie jest kwadratowa rzuć wyjątkiem `ValueError`
- (2) Używając rozwinięcia Laplace'a oblicz rekurencyjnie wyznacznik mat

Dane testowe w skrypcie!

# Zadanie – funkcja obliczająca wyznacznik macierzy

Używając podejścia TDD zaimplementuj:

```
import pytest
```

```
@pytest.mark.parametrize(
    "matrix, expected_determinant",
    [
        ([[5]], 5),
        ([[4, 6], [3, 8]], 14),
        ([[2, 4, 2], [3, 1, 1], [1, 2, 0]], 10),
        ([[6, 1, 1], [4, -2, 5], [2, 8, 7]], -306),
        ([[2, 4, -3], [1, 8, 7], [2, 3, 5]], 113),
        ([[1, 2, 3, 4], [5, 0, 2, 8], [3, 5, 6, 7], [2, 5, 3, 1]], 24),
    ]
)
def test_matrix_determinant_correct(matrix, expected_determinant):
    assert expected_determinant == matrix_determinant(matrix), \
        "Incorrect result!"
```

# Zadanie – funkcja obliczająca wyznacznik macierzy

Używając podejścia TDD zaimplementuj:

```
import pytest

@pytest.mark.parametrize(
    "incorrect_matrix",
    [
        ([[]]),
        ([[4, 6]]),
        ([[2, 4, -3], [1, 8, 7], [2, 5]]),
        ([[1, 2, 3, 4], [0, 2, 8], [3, 5, 6, 7], [2, 3, 1]]),
    ]
)
def test_matrix_determinant_incorrect(incorrect_matrix):
    with pytest.raises(ValueError) as excinfo:
        matrix_determinant(incorrect_matrix)
    assert "Matrix is not square!" in str(excinfo.value)
```

# Zadanie – funkcja obliczająca wyznacznik macierzy

Jak wywołać testy?

```
pip install pytest
```

```
pytest -v <ścieżka do pliku, np. lab_2/matrix_determinant.py>
```