Introduction to Computer Science for Engineers

# Adjacency Matrix - Operations ✔✔✔ ☺

This assignment **closed** January 12, 2025 at 23:15.

In the following we will consider adjacency matrices of the type `list[list[int]] adj`. Node indexing starts with `0`. The *name of a node* is its *index*. Implement the following methods for these matrices:

- A method to determine in *in degree* of a given node `k`. The *in degree* of a node is the number of nodes that are directly connected to node `k`.

  ```
  def in_degree(k: int, m: list[list[int]]) -> int
  ```

- A method to determine the *out degree* of a given node `k`. The *out degree* is the number of nodes that are directly connected from node `k`.

  ```
  def out_degree(k: int, m: list[list[int]]) -> int
  ```

- A method that lists all *adjacent* (directly reachable) nodes of a given node `k`, using the method

  ```
  def adjacent(k: int, m: list[list[int]]) -> list[int]
  ```

- In a graph a **cycle of length 3** is called a *triangle*.
  - Check if the graph given by the following adjacency matrix contains any triangles, and if so, which!

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

  - Write a Python method

    ```
    def has_triangle(m: list[list[int]]) -> bool
    ```

    in which you decide for a given adjacency matrix whether it contains a triangle or not! You may assume that the graph does not contain any edges where the start and end vertex are the same, i.e. `adj[i][i] == 0` for all valid values of `i`.

- Test all your method with suitable examples in the script!

## Template files

Get all files in an archive templates.zip or templates.tgz.

`adjacence_matrix.py`

**Miit Dholakia** | | | | |