Graph Based Pattern Recognition

Exercise 0

Basis

- Chapter 1

Submission

- The submission takes place online on ILIAS.
- Solutions to the theory tasks must be submitted as `*.pdf` file. Other formats will not be accepted.
- Source code for the implementation tasks must be submitted as `*.py` files. Source code that cannot be executed will not be accepted.
- Individual submissions or submissions in teams of two are allowed (hand in only one copy per group). In the source code file, include the *names and matriculation numbers* of both group members in the first two lines as comments.

Dates

- Briefing: 22.02.2023
- Submission: 01.03.2023
- Debriefing: 01.03.2023

Setup Task

Before we start with the first exercises, we will set up a virtual environment in Python and install the necessary packages for the lecture[1]. Creating a virtual environment in Python is best practice to manage dependencies for different projects. This can be done using the *venv* module that is included in Python 3.3 and later versions. Go to the ILIAS's webpage of the course and download/unzip `PR_Lecture.zip`.

To create a virtual environment named *myenv* in the `PR_lecture` directory, you can use the following command:

```
$ cd PR_lecture
$ python3.9 -m venv myenv
```

This will create a new directory called *myenv* containing the necessary files for the virtual environment.

To activate the environment:

```
$ source myenv/bin/activate
```

To deactivate the environment:

```
$ deactivate
```

[1] The exercises have been tested on a Linux operating system using Python version 3.9.

Once the virtual environment is created and activated, you can install the packages listed in the `requirements.txt` file by using the following command:

```
$ pip install -r requirements.txt
```

Implementation Tasks

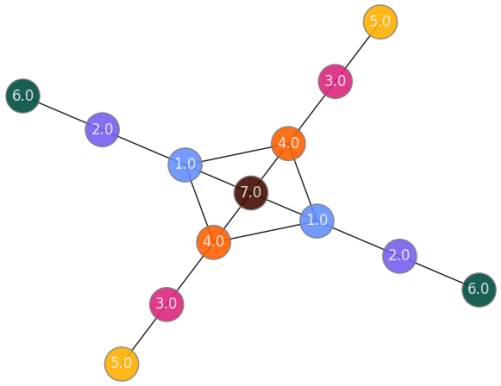
1. The first exercise will familiarize you with the *NetworkX* package. First, go to the ILIAS's webpage of the course and download/unzip `Exercise_0.zip` in your `PR_Lecture` folder. You will obtain the following structure (please do not take into account the structure presented in the video recording):

```
PR_lecture
├── Exercise_0
│   ├── drawings
│   ├── ex0.py
│   ├── graphs
│   │   ├── graph_00.graphml
│   │   ├── graph_01.graphml
│   │   ├── graph_02.graphml
│   │   ├── graph_03.graphml
│   │   └── graph_04.graphml
│   ├── __init__.py
│   ├── results
│   │   └── ground_truth.csv
│   └── utils.py
└── requirements.txt

5 directories, 10 files
```

   Once downloaded/unziped, fill in the necessary code in `PR_lecture/Exercise_0/ex0.py` and `PR_lecture/Exercise_0/utils.py` to load and draw all the graphs contained in the `PR_lecture/Exercise_0/graphs` directory.

   The graph drawings should look similar to the following figure:



   Each node has a label, referred to as *node data* and the color of the node should match the corresponding *node data*.

Hint: You can use the color palette in `PR_lecture/Exercise_0/utils.py`.

2. In the second exercise, the goal is to use the previously loaded graphs and implement a naive graph isomorphism test. To this end, navigate to `PR_lecture/Exercise_0/ex0.py` and complement the second part of the source code.

   Your graph isomorphism test should compare the number of nodes, the number of edges, and the node labels of the two given graphs. If all values of both graphs are identical, your naive test assumes that both graphs are actually isomorphic.

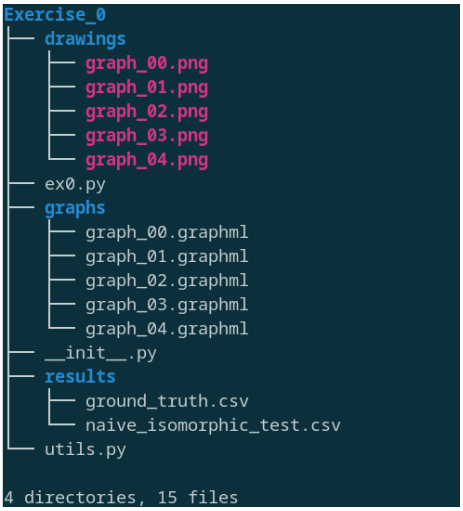
   Create an $N \times N$ binary matrix $\mathbf{R} = (r_{ij})$, where $N = 5$ is the number of graphs in the `PR_lecture/Exercise_0/graphs` directory. Matrix $\mathbf{R}$ contains the results of your isomorphism test for all pairs of graphs. More formally, entry r_{ij} at position i, j is 1, if graph g_i is isomorphic to graph g_j according to your test. Vice versa, entry r_{ij} equals 0, if the corresponding graphs are not isomorphic.

   Save your matrix as `naive_isomorphic_test.csv`

   Compare the results obtained from the naive graph isomorphism test with the ground truth isomorphism (`PR_lecture/Exercise_0/results/ground_truth.csv`) and write a brief explanation for any differences observed. Specifically, when the naive graph isomorphism test fails, explain why the test failed.

   Hint: You can use the graph drawings of the first exercise to compare the results of the naive graph isomorphism test with the ground truth values.

Submission: You must submit a `.zip` file containing the following files.



Additionally, you must submit a `*.pdf` (in the same `.zip` file) that contains explanations of why the naive graph isomorphism failed in some specific cases.
