Exercise Sheet 1: Getting to Know Graphs and the Resource Description Framework

Maximilian Marx, Markus Krötzsch

Knowledge Graphs, 2022-10-25, Winter Term 2022/2023

Exercise 1.1. Show that the number of vertices of odd degree is even in every simple graph.

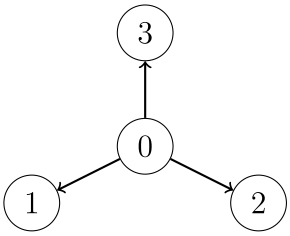
Exercise 1.2. Write a program that reads a directed graph from a file in the format of Exercise 0.3 (recall that test data files are available[1]) and prints out the graph in METIS graph format:

$$\begin{array}{lllll} n & m & & & \\ v_1 & n_1^1 & n_1^2 & \cdots & n_1^{d_1} \\ v_2 & n_2^1 & n_2^2 & \cdots & n_2^{d_2} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ v_n & n_n^1 & n_n^2 & \cdots & n_n^{d_n} \end{array}$$

The first line consists of two integers n and m, separated by a space, where n is the number of vertices, and m is the total number of edges. Each of the following lines specifies the neighbours $n_i^1, n_i^2, \dots, n_i^{d_i}$ of vertex v_i.

As an example, the directed star S_3 would be encoded as:

```
4 3
0 1 2 3
```



Exercise 1.3. A triangle in a directed graph is a simple directed path $v_1 \xrightarrow{e_1} v_2 \xrightarrow{e_2} v_3 \xrightarrow{e_3} v_1$.

Write a program that reads a directed graph G from a file in the format of Exercise 0.3 (cf. the test data files[1]) and prints out the number of triangles in G. How does the runtime of your program scale with the size of the input graph?

Exercise 1.4. A *bipartite graph* is a simple graph $G = \langle V, E \rangle$, where V can be partitioned into two sets X, Y (i.e., $X \cup Y = V$, and $X \cap Y = \varnothing$), such that every edge $\{ a, b \} \in E$ coincides with both X and Y, i.e., $\{ a, b \} \cap X \neq \varnothing$ and $\{ a, b \} \cap Y \neq \varnothing$.

Show that the following are equivalent:

1. $G = \langle V, E \rangle$ is bipartite.
2. G is *2-colourable*, i.e., there is a map $c : V \to \{ 0, 1 \}$ such that no two adjacent vertices a, b have the same colour, i.e., $c(a) \neq c(b)$ for all $\{ a, b \} \in E$.
3. G does not contain a cycle $v_1 \xrightarrow{e_1} v_2 \xrightarrow{e_2} \cdots \xrightarrow{e_{n-1}} v_n \xrightarrow{e_n} v_1$ of odd length.

[1] https://github.com/knowsys/Course-Knowledge-Graphs/tree/master/data/simple-graphs

Exercise 1.5. Write a program that reads a graph in N-Triples format and checks whether the graph is bipartite. Use it to decide whether `authorship.nt.gz`[2] and `coauthors.nt.gz`[2] are bipartite.

Hint: each of the uncompressed graphs is roughly 4 GiB in size. In Python, you can use `gzip.GzipFile`[3] to process the compressed file without decompressing it first. There is also `authorship-snippet.nt.gz`[2], a small part of the graph that you can use during development.

Please note: In order to get the correct data files, please install git-lfs[4] on your system, and then activate it in your local repository (git lfs install).

Exercise 1.6. From the `coauthors.nt.gz` graph[2], extract the *connected component* containing <http://dblp.uni-trier.de/pers/s/Studer:Rudi>, i.e, extract the induced subgraph that

- contains <http://dblp.uni-trier.de/pers/s/Studer:Rudi>,
- contains all nodes reachable from <http://dblp.uni-trier.de/pers/s/Studer:Rudi> by some path, and
- contains all edges that are present in the full graph between these nodes.

Note that, while an RDF graph is inherently directed, edges in `coauthors.nt.gz` are symmetric, i.e., the graph is essentially undirected.
Hint: `authorship-snippet.nt.gz`[2] contains <http://dblp.uni-trier.de/pers/s/Studer:Rudi> and can be used for testing during development.

[2] https://github.com/knowsys/Course-Knowledge-Graphs/tree/master/data/dblp

[3] https://docs.python.org/3/library/gzip.html

[4] https://git-lfs.github.com/