# Master 2 Science des données - A2IA
# Machine Learning on Graphs
# Practical lab - session 1

S.Nicolas

September, 2025

**Goal :** in this first practical lab we propose to use the NetworkX library in order to grasp the general notions of graph theory seen in the first lesson, and to study the properties of different types of graphs such as simple graphs, multigraphs, directed and undirected graphs, weighted graphs.

**Instructions :** during the session, you will carry out the compulsory work requested below. You should preferably use a Python notebook for your implementation. You must submit your work at the end of the session, on the UniversiTICE deposit space provided for this purpose. The optional work proposed in the "additional work" section may be carried out or completed outside the practical sessions, and must be handed in by the end of the semester at the latest. This work is optional but will be awarded additional points if it is completed.

## 1 First part (mandatory) : getting started with NetworkX

In this first part you will use NetworkX which is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks. You can access the documentation for the latest stable version of NetworkX here : `https://networkx.org/documentation/stable/index.html`

As an application case, we will consider the game boards of two board games which use graph representation : the games "Ticket to Ride" and "Pandemic". In fact network structures are at the heart of gameplay in many games. So we will use NetworkX to implement the underlying network structure represented by these two game boards, and the functions offered by NetworkX to characterise and study the properties of these graphs. There are many versions and extensions of these games, but here we have chosen two versions for which the network represented on the game board is of a reasonable size given the time you have available for practical work, namely "Ticket to Ride : London" and "Pandemic Hot Zone USA".

In this work, we are not interested in modelling games, so we will not describe the aim, rules or victory conditions of these games. We will simply look at the game board and how it is modelled in the form of a graph.

The "Ticket to ride : London" game board, illustrated by the figure in the attached document, is a simplified map of London, with the city's main centres of interest linked by public transport lines (underground, tram, train, bus, etc.). Roughly speaking, the length of each line is defined by the number of segments that make it up. For example, the line between "Baker Street" and "British Museum" is made up of 4 segments, so has a value or length of 4. You can see on the board that the lines are coloured, mainly for visibility reasons, but these colours are not taken into account in the rules. The nodes are also coloured and numbered.

The "Pandemic Hot Zone USA" game board represents a simplified map of the United States, on which the main towns are linked by communication routes (air or land transport systems), encouraging the spread of a virus between towns. In this game, links between towns are not characterised by length or any other attribute. All that matters is the presence or absence of a link. Cities are simply associated with a colour that reflects their geographical distribution (southern, eastern or western cities).

**Required work :**

1. Using NetworkX and the appropriate graph structure, implement the underlying graph structure represented by these two game boards, taking into account the attributes of the nodes and edges ? What types of graphs are involved ?
2. Determine the order and size of these graphs
3. Show both graphs as adjacency lists
4. Determine and display the adjacency, incidence and Laplacian matrices of these graphs
5. Using the definition of the degree matrix and the relationships with the other matrices seen in class, print the degree matrix for the board of "Pandemic" game
6. Determine and print the minimum and maximum degree, as well as the average degree
7. Display the eccentricity of all the nodes of these graphs

8. Display the diameter, the radius and the girth of both graphs
9. Using the appropriate NetworkX function, determine whether these graphs are connected or not, and display the answer
10. Compute the density of these graphs and verify using NetworkX
11. List the nodes of these graphs in descending order of degree. Do the same for the centrality values
12. Determine the distance between Santo Domingo and Calgary on the "Pandemic" map, and find the most distant pairs of nodes
13. Determine whether these graphs are Eulerian or not, and display the number of possible Eulerian paths if there are any

NetworkX is not a package dedicated to network visualization, but it nonetheless includes display functions. Display both graphs by trying out the various proposed drawing functions. Note that some other Python packages or libraries are dedicated to network drawing, such as Pyvis (`https://pyvis.readthedocs.io/en/latest/index.html`)for example. If you have the time, you can experiment network drawing using this package.

As you can see, the two graphs studied above are undirected graphs. So now create a directed graph structure of your choice, and test the various functions used previously on this graph. An example of a directed graph is the network we obtain if we model the relationships that exist in certain networks such as X (formerly Twitter), in which the "follow" relationnish are directed.

## 2 Second part (optional) : implementation of a Graph class

In this optional part, you will implement in Python a graph data structure (Graph class) allowing to manipulate simple undirected weighted or unweighted graphs (node/edge addition and deletion) and providing information on graphs (order, size, adjacency and degree matrices,...)