Assignment #2

- Show your understanding of the basic practical tools needed to analyze a given network, using **Python** (or, alternatively, R/C/Java or any other language you feel comfortable with);
- You are likely to include/install many libraries. For instance, **networkx** is probably the best choice because we used it during our Lab sessions. Also **igraph** worths a look;
- **gephi** is a great tool if you need to analyze/visualize an adequately small network. You can use gephi at this stage to better and quickly visualize the network, but you must focus on other tools to perform the majority of your measures (see next slide).

Objectives

- You are requested to give a **generic description** of the network. Calculate the measures and the characteristics below, and plot them whenever it is possible/significant:
  - **Distances**: average, distribution
  - **Degree**: average, variance/standard deviation, degree distribution (some fit? Does it follow a power law? If yes, is it in the scale-free regime?)
  - **Clustering** coefficient
  - **Largest connected component** size
  - **Degree correlation**: neutral, assortative, disassortative?
  - Are there **communities**? Can you properly show them with an appropriate layout? Can you discuss them?
  - **Centralities**
  - Can you analyze **homophily**?
  - …
- Try to interpret the results of these measures, and comment/discuss results.

Random models (not mandatory)

- You probably need to create your own **artificial random networks** for comparison purposes
- You can use **different generative models**, to produce comparisons, varying some parameters (e.g., linking probability, number of edges added at each step, degree distributions, and so on)
- For some comparisons, you may need to preserve some characteristics (e.g., degree distribution). Try to **rewire** properly your network in order to shuffle your data.
- You can perform on such synthetic networks the **analysis** that has been proposed in the previous slides, to detect differences
- Try to explain **different behaviors**
- You can get **inspiration** from some of the exercises you solved in the theoretical part of the previous assignment

Datasets

- **Hint**: before challenging yourself with a “serious” dataset, try to execute your assignment with a smaller one. You can find many of them around
- Once you have developed your code and trained your skills, attack one of the datasets you can find in one of these sources:
  - datasets attached to notebooks we studied during our course (you can expand the analysis we have already studied together)
  - SNAP: https://snap.stanford.edu/data/index.html
  - KONECT: http://konect.uni-koblenz.de
- Make your own selection!
- If previous studies on the dataset you selected are available, use them as a reference or for comparison
- Warning: avoid large datasets (you may suffer of a lack of computational power…)

Select your partner(s)!

- You can work **alone** or with (an)other **partner(s) - max team size: 3 members**
- Let's exploit **heterophily**: If you think you are strong with the mathematical principles, but weak on programming (or vice-versa), try to join a partner that completes your skills
- **Deadline**: Just remember to submit on moodle your technical report at **least three days** before the final exam. Use the same folder on moodle to submit your reports (zip them to upload only one file)
- **Recommendation**: use Latex, adopting an **article template**, and listing the names of the authors