

# **Algorithms and Applications in Social Networks**

## **HW #1**

**Instructions:** Implementation should be done using Python and NetworkX library. Please submit your code in .py files (file per question) or .ipynb file (Jupyter Notebooks).

The theoretical part of the question should be submitted in PDF file.

Do not forget to write IDs of all member in the team (pair). Submit only once per team!

Please ZIP all files together, name the file HW1\_<student\_id>.zip and upload it to Moodle.

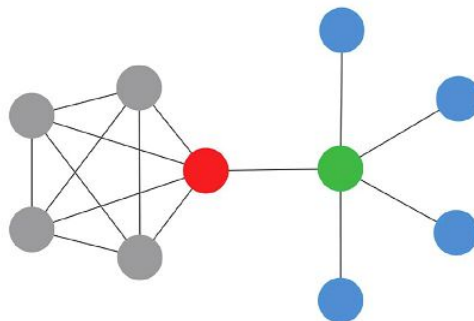
### **Question #1:**

- Implement an Erdős–Rényi model. Write a function that gets  $n$  - number of nodes and  $p$  - probability of an edge and returns random graph.
- Implement Small World mode. Write a function that gets  $n$  - number of nodes,  $k$  - average degree and  $p$  - probability of recreation of an edge and returns random graph.
- Implement function that computes the clustering coefficient of a given graph. Implement a helper function that computes clustering coefficient of each node.
- Run the two implemented functions: Erdős–Rényi with  $n=10000$ ,  $p=0.2$  and Small World with  $n=10000$ ,  $k=8$ ,  $p=0.1$

Analyze the results by comparing the degree distribution, the diameter and the clustering coefficient of the graphs

### **Question #2:**

- Implement 3 centrality measures: Degree centrality, Betweenness centrality and Closeness centrality.
- Run these measures on Florence families network and report top-5 nodes according to each of the measures.
- Visualize the network using different centrality measures by changing node sizes based on the centrality measure value of each node.
- Compute (manually) each of the centrality measures for red, green, blue and grey nodes. Rank the nodes by each of the measures and explain the difference between gray and green nodes in terms of different centrality measures.



**Question #3:**

- a. Implement `check_balance(G)` function that checks balance (according to “Theory of Structural Balance”) of a  $G$ . Explain your implementation decision.
- b. Generate a random network using Erdős–Rényi using  $n=30$  and  $p=0.5$ . On each edge, assign random sign (“+” or “-”) with different probabilities:
  - i.  $p(+) = 0.9, p(-) = 0.1$
  - ii.  $p(+) = p(-) = 0.5$
- c. Run the `check_balance(G)` on each of the graphs. Visualize the results (i.e. highlight the reason if the graph is not balanced).
- d. Think about real-world example of signed social network (up to 20 nodes). Present the network, explain the nodes and the edges. Check (manually) if the network follows the “Theory of Structural Balance” and explain the results.