**Exercise 3 – Final Project**

The election manipulation algorithm is mainly divided into two files, manipulator.py and selection.py.  
In manipulation.py there is the manipulation(G, p, c, B, b) function as required, together with the plurality\_voting\_rule(p, b) function, which counts the votes for each candidate according to the plurality voting rule, and to the FriedkinJohnsen(G, stubborness, belief) function which performs the FJ dynamics on graph G according to the stubborness and belief passed in input.  
In selection.py there is the selector(G, B) function which returns the best B seeds of graph G based on the linear combination of the Shapley values ​​based on the Shapley Degree, Shapley Threshold (imported from the shapley.py file in the exercise1\_final folder) and Shapley Closeness. The Shapley Closeness algorithm has been included in the selection.py file because a parallel version has been implemented to minimize execution times.  
Therefore, in order to execute the manipulation function, it is necessary to be able to import the functions present in selection.py and exercise1\_final/shapley.py.

The various experiments were carried out on different networks (all the networks provided for exercise 2 of the final project and other networks recovered from the internet) and it was possible to experimentally determine how the game-theoretic centrality measures allow obtaining better results than the classic ones. Despite this, each of the three types of Shapley value obtained low results on different types of networks, such as the Shapley Closeness on the net\_3 network. On the contrary, their linear combination did not show this type of problem, performing well on every network, in some cases improving them and in others getting very close to the results of the best Shapley Value for that network.  
Finally, different k values ​​were tested in the Shapley Threshold, concluding that too high values ​​caused a worsening of the results and low values ​​tended to obtain the same results as the Shapley Degree and did not bring improvements, therefore choosing to use a k equal to 5.

In addition, clustering algorithms were also tested in the various experiments to select the seeds in each cluster in proportion to their size. However, this choice did not lead to improvements, therefore deciding not to include it in the final solution to reduce execution times.