**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.