Task1: Social Network Mining

Implement, if necessary, optimized versions of the social network mining algorithms seen during the course (*diameter*, *triangles computation*, *clustering*) and test these algorithms on the following datasets:

- **Facebook Large Page-Page Network** available at https://snap.stanford.edu/data/facebook-large-page-page-network.html
- **High-energy physics theory citation network** available at https://snap.stanford.edu/data/cit-HepTh.html

Note that the first network is undirected, while the latter is directed (so you need to adapt algorithms to work with both the types of networks).

The goal of this task is to collect some algorithms that are able to perform efficiently on the provided networks, by guaranteeing quite precise results. Indeed, in the final project you will be required to mine a network similar to the provided ones. This task has thus the goal to shortlist the set of algorithms that you will use in the final task.

NOTE: In order to evaluate the precision of collected algorithms, you need to compare the optimized algorithms against correct algorithms. The latter may have excessively long running time on the provided networks. If this is the case you can compare the precision of your optimized algorithms with respect to smaller networks available at the following repositories: **SNAP** (https://snap.stanford.edu/data/), **KONECT** (http://konect.cc/networks/), **networkrepository.com** (https://networkrepository.com/index.php).

Task2: Centrality Measures

Implement the *shapley-closeness* centrality measure as defined in Michalack et al. (JAIR 2013) sec. 4.4.

Implement, if necessary, optimized versions of all studied centrality measures (*degree*, *closeness*, *betweenness*, *PageRank*, *HITS-authority*, *HITS-hubiness*, *HITS-both*, *voterank*, *shapley-degree*, *shapley-threshold*, *shapley-closeness*) and test them on the datasets indicated in Task1.

The goal of this task is to shortlists the set of centrality measures based on efficiency and similarity of outcomes. Indeed, in the final project you may need to use centrality measures. This task has the goal to shortlist the set of measures that you will use in the final task.

Task3: Social Network Mechanisms

Implement the **VCG**, the **MUDAN**, and the **MUDAR**, for selling multiple homogeneous items on a social network, with each agent only requiring a single item. The **MUDAN** and **MUDAR** algorithm are available on (Fang et al., 2023).

Each mechanism must be implemented through a function
auction(k, seller_net, reports, bids), where

- **k** is the number of item to sell;
- **seller_net** is a set of strings each identifying a different bidder;
- **reports** is a dictionary whose keys are strings each identifying a different bidder and whose values are sets of strings representing the set of bidders to which the bidder identified by the key reports the information about the auction;
- **bids** is a dictionary whose keys are strings each identifying a different bidder and whose values are numbers defining the bid of the bidder identified by that key.

You may assume that the input is well-formed, i.e., **bids** contains an entry for *all* and *only* the bidders that are either in **seller_net** or in the report of some other bidder; and **reports** contains only entries for bidders that are either in **seller_net** or in the report of some other bidder. Note that it may be the case that some bidders appear as keys in the **bids** dictionary, but they do not as keys in the **reports** dictionary: this must be interpreted as these bidders do not reporting the information to any of their neighbors.

The function returns two values:

- **allocation**, that is a dictionary that has as keys the strings identifying each of the bidders that submitted a bid, and as value a boolean **True** if this bidder is allocated one of the items, and **False** otherwise.
- **payments**, that is a dictionary that has as keys the strings identifying each of the bidders that submitted a bid, and as value the price that she pays. Here, a positive price means that the bidder is paying to the seller, while a negative price means that the seller is paying to the bidder.

Compare the performance of these algorithms in terms of running time, of seller revenue, and of social welfare, on the networks described above with

- different values of **k**;
- different choices for the seller as a randomly chosen node of the graph;
- different choices for the valuations to be randomly selected in **{1, 2, …, 10000}**;
- bids and reports to be truthful.

For each of the three auction formats, evaluate if truthful bidding was an equilibrium (i.e., does an agent prefer to submit an untruthful bid if the remaining agents are truthful?), and, if not, estimate how often truthful bidding fails to be an equilibrium.

The problem faced in this task is a simplification of the one faced in the final project. Hence, this task can allow to shortlist diffusion auction formats that may have chances to achieve good performances even in the final project.

Task4: Learning

Implement an environment for bandits algorithm that will model the following setting:

> We are given a social network such that with a probability **p(u,v)** for each edge **(u,v)**. These probabilities are unknown to the learner. The learner at each time steps interacts with this environment by choosing a vertex **x**. The environment set each edge **(u,v)** as *alive* with probability p(u,v), and *dead* otherwise. It then assigns as a reward to the learner that is equivalent to the number of nodes of the social network that are reachable from the selected vertex **x** through alive edges only.

Given that the goal of the auctioneer is to maximize her cumulative reward, find the best bandit algorithm (either found among the ones implemented during the course, or a completely new designed algorithm) that the auctioneer may use in above setting (i.e., the algorithm must decide which vertex to select at each time step).

The problem faced in this task is a simplification of the one faced in the final project. Hence, this task can allow to shortlist these bandit algorithms that may have chances to achieve good performances even in the final project.