Diffusion Processes on Complex Networks - Lab

Assignment 2

Janusz Szwabiński

1. Write your own implementation of the graph structure in the language of your preference. It should have the following interface:

   - `Graph()` creates a new, empty undirected graph.
   - `addVertex(vert)` adds a vertex to the graph.
   - `addVerticesFromList(vertList)` adds a list of vertices to the graph.
   - `addEdge(fromVert, toVert)` adds a new, undirected edge to the graph that connects two vertices.
   - `addEdge(fromVert, toVert, weight)` adds a new, weighted, undirected edge to the graph that connects two vertices. If not in the graph, the vertices should be added automatically.
   - `addEdgesFromList(edgeList)` adds a list of edges to the graph.
   - `getVertices()` returns the list of all vertices in the graph.
   - `getEdges()` returns the list of all edges in the graph.
   - `getNeighbors(vertKey)` returns the list of all neighbors of the vertex labeled `vertKey`.
   - `in` returns True for a statement of the form vertex in graph, if the given vertex is in the graph, False otherwise.
   - `saveGraph(graph)` writes `dot` representation of the graph to a text file

   To check your implementation, take the list of edges given in Assignment one, build the corresponding graph, save it to text file and then draw it by making use of the `graphviz` software (you may look at `http://www.webgraphviz.com/` in case graphviz is not installed). Compare the resulting graph with the one generated in `Gephi`.

2. Add the following method to the above implementation:

   - `getShortestPaths(fromVert)` calculates shortest paths in the graph from the given vertex to all other vertices.

   Use the already known social network to check how the method works.