

## Assignment 1

Date of issue: 28.02.2019

Rana Hussein: [rana.hussein@unifr.ch](mailto:rana.hussein@unifr.ch)

### Graph Representation and Traversal

**Task 1.** Consider a webgraph representing how websites are connected, where nodes represent URLs and edges represent hyperlinks between them. Answer the following questions:

1. Compute the time complexity of finding whether two URLs are connected or not, for adjacency matrix, adjacency list and edge list.
2. What is the most time efficient graph representation to find the number of links for a particular URL in the network.
3. Explain how to use the adjacency matrix representation to find out if the graph is directed or undirected.

**Task 2.** Create a Python script using NetworkX python library and perform the following tasks:

1. Download and extract the social network *CollegeMsg.txt.gz* from <http://snap.stanford.edu/data/CollegeMsg.html>. The dataset represents an online social network of the University of California. Nodes represent users and edges represent messages between them.
2. Load the network to your script and print the network information (such as number of nodes and edges) using *NetworkX* library.
3. Perform a Depth-First-Search (DFS) traversal starting from Node 1 using *NetworkX* library, and list the order of traversed edges.

Library details:

- NetworkX: Python language software package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks. Download and install NetworkX library from <https://networkx.github.io/>

**Task 3.** Create a Python script using Matplotlib python library to plot the college social network from Task 2. What do you notice about the structure of the network?

Library details:

- Matplotlib: Plotting library for the Python programming language. Download and install Matplotlib library from <http://matplotlib.org/>

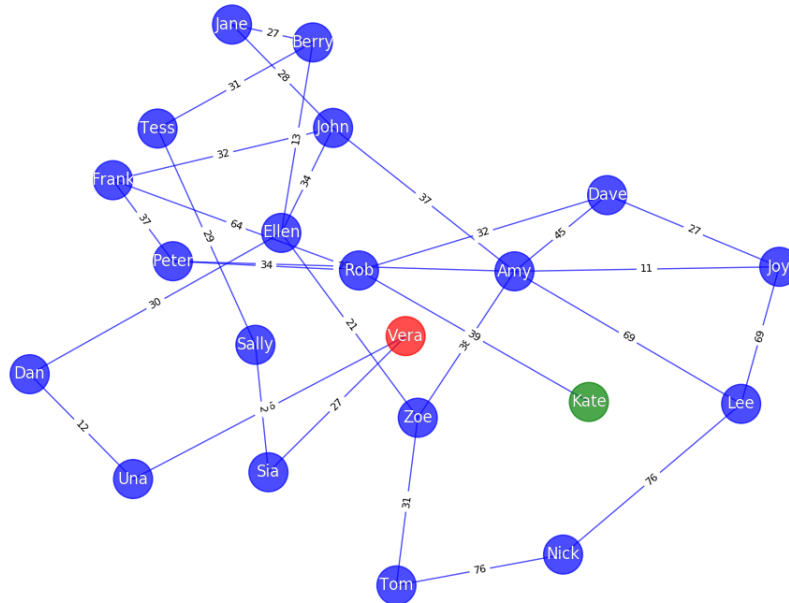


Figure 1

**Task 4.** Given a social media graph, where nodes represent users and edges represent friendships between them as shown in Figure 1. Consider the following costs functions to propagate a piece of information across the network:

- **Cost to come:** In (file: *users\_edgelist.txt*), each line is in the form: *user1 user2 cost*, which represents an edge between *user1* and *user2*.
- **Cost to go:** In (file: *users\_heuristics.txt*), each line is in the form: *user cost*, which represents the cost from each user to destination user: *Vera*.

Given the incomplete Greedy Search algorithm implementation (file: *greedy\_search.py*), perform the following tasks:

1. Load the network (file: *users\_edgelist.txt*) and heuristics into your script.
2. Complete the Greedy Search algorithm implementation, to find the best route to propagate a piece of information starting from user: *Kate* to user: *Vera*.
3. Visualize the network and highlight the greedy path using Matplotlib Library.

**Assignment delivery notes:**

- This assignment should be uploaded to ILIAS by 06.03.2019
- Tasks 1 should be uploaded in *pdf* format.
- Tasks 2-4 should be uploaded in *py* format.