FRENCH-AZERBAIJANI UNIVERSITY
UE709 Network Algorithms L3/S5
PW4 - Network and Algorithms
November 2019

Working with Python and NetworkX (cont.3)

1 Requirements

This practice (continuation of the last practice) uses the source file produced in last Practical Works. Section 2 gather some information needed to this practice.

2 More information on NetworkX

i. Start as usual, importing modules in need and recovering the network graph from file:

```
import pandas as pd
import networkx as nx

data = pd.read_csv('sample_network.csv')
G = nx.from_pandas_edgelist(data, source='Source', target='Sink',
    edge_attr=True, create_using=nx.Graph)
```

where `Source` and `Sink` are the titles (names) of columns that will be considered as the endpoints of each edge (row). Other columns are data for each edge (`edge_attr=True`). Note that we are **using the `create_using=nx.Graph` option** to create our graph.

iii. About nodes and edges:

- `G.nodes()` returns a list with every node of the graph G
- `G.nodes[node_identifier]` returns a dictionary with all attributes for the `node_identifier` of G
- `G.nodes[node_identifier][attribute]` returns a dictionary with all attributes for that node
- `G.edges()` returns a list of tuples constituted by the endpoints of all unique edges (undirected) of G
- `G.edges[node_id1, node_id2]` returns a dictionary with all attributes for the edge `(node_id1, node_id2)` of G

- `G.edges[node_id1, node_id2][attribute]` returns the value of the attribute for that edge `(node_id1, node_id2)`
- `list(G.neighbors(node_identifier))` returns a list with all nodes that share an edge with the `node_identifier` (don't need to convert to list if you need to iterate through it)
- `G.degree(node_identifier)` gives the degree of a node

v. To find more information about NetworkX:
https://networkx.github.io/documentation/latest/index.html

3 Activities

3.1 Network flow

The `sample_network.csv` dataset have a sample of nodes with a fictional average rate Kbps for communication between each pair (edge's attribute).

1. Plot the network.
2. Use the following metrics to help you understand the network in analysis:
   - Degree Connectivity
   - Closeness Centrality
   - Betweenness Centrality
   - Network Density
   - Network Diameter
   - Network Average Path Length
3. Implement **your version of <u>one</u> of the following algorithms**:
   - The Ford-Fulkerson algorithm
     https://en.wikipedia.org/wiki/Ford%E2%80%93Fulkerson_algorithm
   - The Edmonds-karp algorithm
     https://en.wikipedia.org/wiki/Edmonds%E2%80%93Karp_algorithm
   - The Dinic's algorithm
     https://en.wikipedia.org/wiki/Dinic%27s_algorithm
4. Find the max flow of information for this network.