INSTITUTO SUPERIOR TÉCNICO | UNIVERSIDADE DE LISBOA
NETWORK SCIENCE 2021/22
Problem Set 1

1. Time and space complexity[1]

1.1. Fill the following table about graph operations complexity (worst case):

	Edge vector	Adjacency matrix	Adjacency list
Space	O( )	O( )	O( )
Initialization	O( )	O( )	O( )
Insertion of an edge	O( )	O( )	O( )
Find an edge	O( )	O( )	O( )
Remove an edge	O( )	O( )	O( )

Note: Please check class 02 and assume that the graph has N vertices and E edges.

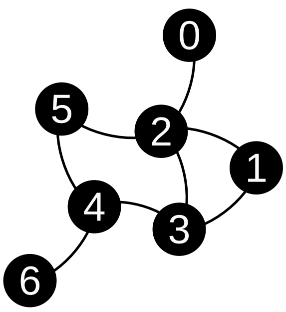
1.2. What is the time complexity, as a function of the number of vertices N and E edges, of finding the highest-degree vertex in an undirected network, assuming the vertices are given to you in no particular order? Please assume that the network in question is stored in adjacency matrix format.

1.3. Assuming adjacency lists, assess the time complexity associated with the computation of the degree distribution of an undirected network with an average degree $z = \frac{2E}{N} << (N-1)$.

1.4. For an undirected network of N vertices and E edges stored in a adjacency list format, assess the complexity associated with the computation of the diameter of a network.

1.5. For an undirected network of N vertices and an average degree of *z*, discuss the average time complexity associated *i)* with listing all first neighbors of a node and *ii)* with listing all second neighbors of a node.

2. Consider the following network (if you feel like playing with a real data-set, please try ex. 3).



2.1. Show its adjacency matrix and its adjacency list.
2.2. Find the degree for each vertex.
2.3. Compute a BFS starting at vertex 0 and show the values for *d* (discovery time or distance).
2.4. Compute the clustering coefficient and closeness centrality for vertices 0 and 2.
2.5. (*facultative*) An alternative form of computing the global clustering coefficient (or transititivity) common in social sciences is based on counting the number of connected triplets. This definition assumes that the clustering coefficient is given by

$$C = \frac{3 \times \text{number of triangles}}{\text{number of connected triplets}}$$

where a connected triplet is defined to be a connected subgraph consisting of three vertices and two edges. The value 3 in the numerator comes from the definition of a closed triplet, which consists of three connected nodes. A triangle therefore includes three closed triplets, one centered on each of the nodes. Discuss the equivalence between the two definitions of clustering. Provide an example if needed to support your conclusion.

[1] Solution: **1.1.** see class 02, **1.2.** O(N²). **1.3.** O(N+N.2E/N+N)=[histogram setup]+[loop over all nodes]+[normalization]=O(N+E). **1.4.** O(N.(N+E)) = N applications of a BFS (N+E).**1.5.** O(z) and O(z²), respectively.

3. Let us play with some tools and data.

3.1. Download and run Gephi[2]. For those that are using this software for the first time, perhaps the fastest way to get into its details is by watching one of the many video tutorials available (check, for instance, this one: http://www.cs.umd.edu/~golbeck/INST633o/Viz.shtml).

3.2. Open Gephi and create the network of exercise 2 (if you prefer to start with a real dataset, go to the next exercise). Confirm the metrics discussed above and explore other metrics available. Use the node's attributes to create colour gradients for degree, clustering and betweeness centrality for this simple network.

3.3. Then load the Dolphin social network in Gephi[3] (Note: if you highlight the attribute "Modularity Class" and adopt a "Layout" based on "Yifan Hu" algorithm you should get the network at the right)[4]. You may also check the original paper where this dataset was proposed[5].

3.4. Compute (using Gephi) the local clustering coefficient, the closeness centrality and the betweenness centrality, for each vertex. Compare and discuss obtained vertex rankings.

3.5. Find the shortest path between *Whitetip* and *Jet*. (use the "airplane tool" on your left).

3.6. Explore the node's attributes in Gephi.

4. Let us continue to play with new tools and data. Start exploring a (more serious) network package associated with your favorite language. This may take a bit. Start as soon as possible. Here are some suggestions:

- Python: If you feel like using Python, we recommend the use of NetworkX. Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks. Here, we sould suggest to follow the notebooks available here: https://github.com/bjedwards/NetworkXTutorial.
- C/C++/R/Python: igraph. collection of network tools with the emphasis on efficiency and portability (can be programmed in R, Python and C/C++).
- C++: Boost library (free peer-reviewed portable C++ source libraries, which also includes a large set of graph algorithms and data structures).
- JAVA: JUNG, Java Universal Network/Graph Framework
- Mathematica & network analysis: Recent versions of Mathematica offer several interesting network science tools. You may also check Wolfram Demonstrations Project, where you can find some examples of notebooks related to complex networks, epidemiology, game theory, and others.

5. Play with other networks available at Network data pointers by Mark Newman[6], or any other network repository (for other suggestions please check our webpage).

---

[2] https://gephi.org/
[3] http://www-personal.umich.edu/~mejn/netdata/dolphins.zip
[4] After adding all nodes and edges you should end up with something like this: https://fenix.tecnico.ulisboa.pt/downloadFile/1689468335593478/dophins.gephi
[5] Lusseau, David. "The emergent properties of a dolphin social network." Proceedings of the Royal Society of London B: Biological Sciences 270.Suppl 2 (2003): S186-S188. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1809954/pdf/14667378.pdf
[6] http://www-personal.umich.edu/~mejn/netdata/