

# Complex Networks Winter School

## Practical session 1

The goal of this practical session is to get familiar with two different frameworks for processing complex networks. The first one will be NetworkX and the second one Giraph. You are free to start with the one you want, there is no dependency.

**Ex. 1** — (NetworkX) For this exercise, the following software and libraries are required : Python 2.7+, NetworkX, matplotlib, numpy, graphviz and pygraphviz. The last 4 packages are required for visualizing the networks.

- 1.Install everything
- 2.Create a graph containing 10 nodes forming a ring and display it
- 3.Remove every odd edge and display the graph

Directed graphs have to be explicitly created as *DiGraph*.

- 4.Create a directed graph containing 10 nodes forming a ring and display it
- 5.Create a binary tree of depth  $n$  and display it

Now that we are familiar with basic graph creation, we can look at I/O.

- 6.Write in a text file an adjacency matrix. For each vertex  $V_i$  line should be in the form  $V_i V_a V_c \dots V_x$  if there exist an edge between  $V_i$  and  $V_a \dots$
- 7.Load the file in NetworkX and draw it (see <http://networkx.github.io/documentation/latest/reference/readwrite.html>)
- 8.Use the graph generators to build a graph and save it on disk

Now we can finally start analyzing the graphs.

- 8.Generate (or read from disk) a graph and compute a Single Source Shortest Path.
- 9.Compute basic properties like min, max, average degree and connected components
- 10.Implements a *Single Source Shortest Path* and compare it with the one provided in NetworkX

11. Do the same with PageRank

**Ex. 2** — (Giraph) For this exercise, the following software are required :  
JDK7, Maven 3.X

1. Download and install the Hadoop 1.0.2 distribution
2. Configure Hadoop to run in *Single Node Cluster* mode
3. Start your Hadoop cluster (bin/start-dfs.sh and bin/start-mapred.sh) and cross your fingers. You can also check that a HTTP server is running on ports 50070 and 50030.
4. Execute a word count to get familiar with the submission system.

The second step is to install Giraph. Unfortunately there is no distribution available so one has to build it from source, using Maven.

5. Download the latest Giraph archive
6. Read the README and compile using Maven. Go drink a coffee while Maven downloads the Internet.
7. If everything is fine, you should have a giraph-xxx.jar which can be submitted as any jar on Hadoop.

It is beyond the scope of this session to actually write a new code, we will simply use an existing example from the distribution.

9. Use the following guide to run the PageRank Benchmark <https://cwiki.apache.org/confluence/display/GIRAPH/Quick+Start+Guide>
10. Modify the benchmark to use a file describing the graph as an input