COMP5048 Week 5 Tutorial

1. yEd

yEd is a graph visualisation software.

1. Go to the yEd website at https://www.yworks.com/yed and go to the download page.
2. Select the appropriate package for your system and download.
3. Run the installer or uncompress the package. For compressed packages, yEd commands can now be executed from the directory where the archive has been uncompressed.

This tutorial will run a few examples to show how to load data, create layout and customize visualization.

Runthrogh Examples

1.1 Load a graph from file

1. Run yEd from the command line
2. Choose Open and then select a file, e.g., select can_144.graphml
3. After the graph is loaded, choose "Organic" layout. A visualization is shown.

1.2 Visualize a tree using yEd

1. Load tree.tgf
2. Choose Radial layout. Then try to use other tree layouts such as Tree (Directed) and Tree (Balloon)

1.3 Visualize general graphs

1. Load or create a random graph
2. Select Circular layout. Select Orthogonal layout. Select Organic layout.

1.4 Visualize directed graphs

1. Load a graph tree.tgf. Then select Hierarchical layout
2. Create a planar graph. Then select Hierarchical layout

1.5. Customize visualization

1. Create a graph and visualize it using Organic layout
2. Select Tools / Analyze Graph. Then see some statistics of the graph
3. Select Centrality Measures. Select 'Number of Connected Edges'. Nodes are then colored and sized based on degree centrality

1.6 Create a visualization of a tree using yEd

1. Create a tree.
2. Then use Tree (Directed), Tree (Balloon) and Radial layout
3. Use degree centrality to color nodes

1.7. Partial layout

1. Load a graph and apply a layout
2. Use mouse and select a region of the graph, then select Layout / Selection (Partial).
3. Select a layout type to use for the selection.

2. Networkx

Networkx is a Python software for analysis of complex networks. Installation instructions are given at https://networkx.github.io/documentation/networkx-1.10/install.html.

1. Make sure to have setuptools installed first
2. Then issue the command: pip install networkx

This tutorial will give several examples to show how to use the software. For more usage examples, see at https://networkx.github.io/documentation/networkx-1.10/tutorial/index.html.

Runthrough examples

1. Make sure to import the networkx package before using it. The following lines will create an empty graph:

   ```
   >>> import networkx as nx
   >>> G=nx.Graph()
   ```

2. To add a node to the graph, use

   ```
   >>> G.add_node(1)
   ```

3. To add an edge, use

   ```
   >>> G.add_edge(1,2)
   ```

4. To get the number of nodes and edges, use G.number_of_nodes() and G.number_of_edges().

5. Analyze a graph: some graph-theoretic functions can be directly invoked. For example, to get the number of connected components of a graph G, use nx.connected_components(G). To get the clustering of a graph G, use nx.clustering(G). Furthermore, nx.degree(G) returns a degree mapping of G.

6. Drawing graphs: NetworkX has basic drawing with Matplotlib and it has an interface to use the Graphviz software package.

   First, import Matplotlib's plot interface: import matplotlib.pyplot as plt.

   To test if the import of networkx.drawing was successful draw G using one of:

   ```
   >>> nx.draw(G)
   >>> nx.draw_random(G)
   >>> nx.draw_circular(G)
   >>> nx.draw_spectral(G)
   ```

   when drawing to an interactive display. Note that you may need to issue a Matplotlib plt.show(). To save drawings to a file, use, for example, plt.savefig("path.png").

For more tutorials and examples, see https://networkx.github.io/documentation/networkx-1.10/examples/index.html. NetworkX comes with many analyses. The reference is available at http://networkx.readthedocs.io/en/stable/reference/index.html.