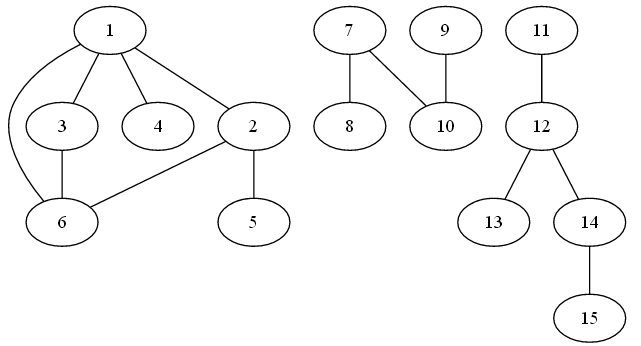
COMP 4030/6030 -- Assignment 2

Due: 9/14/2017 before class

1. (50 Points) Given a directed graph **G** and a pair of nodes **u** and **v**, design a Python program that determines if **u** is reachable from **v**, and **v** is reachable from **u**. You can take advantage of the exercises we did in class.
2. (50 Points) An undirected graph **G** (think of it as a social network) is a graph in which edges are not directed. If **(u,v)** is an edge, then **(v,u)** is also an edge. Here is how you can generate a random undirected graph from the graph module we used:

G = graph.random\_graph(10, 0.4, directed=False)

Write a Python program to count the number of communities in an undirected graph. Two communities, in this problem, are not connected at all. For example, the graph below has 3 communities:



Hint: you can explore undirected graphs the same way you explore directed graphs. Nodes in undirected graphs do not have .In and .Out neighbors. Instead, they only have a .Neighbors attribute. In other words, the ids of neighbors of node 0 are G.Neighbors[0].

**Turn in instructions:**

* The name your solution file should be the same as your UID, plus a .py extension. For example, if your UID is jsmith (i.e. your email is [jsmith@memphis.edu)](mailto:jsmith@memphis.edu)), then your solution file should be **jsmith.py**.
* In the file, put your full name, COMP 4030 or COMP 6030, and Assignment 2.
* Send your solution to the TA (Quang Tran, [qmtran@memphis.edu)](mailto:qmtran@memphis.edu)) with the subject line “**COMP 4030 Assignment 2**”.