Assignment 4

In [1]:
```
import networkx as nx
import pandas as pd
import numpy as np
import pickle
```

Part 1 - Random Graph Identification

For the first part of this assignment you will analyze randomly generated graphs and determine which algorithm created them.

In [2]:
```
P1_Graphs = pickle.load(open('A4_graphs','rb'))
P1_Graphs
```

Out[2]:
```
[<networkx.classes.graph.Graph at 0x7fa0b690e8d0>,
 <networkx.classes.graph.Graph at 0x7fa0b690e9e8>,
 <networkx.classes.graph.Graph at 0x7fa0b690ea20>,
 <networkx.classes.graph.Graph at 0x7fa0b690ea58>,
 <networkx.classes.graph.Graph at 0x7fa0b690ea90>]
```

`P1_Graphs` is a list containing 5 networkx graphs. Each of these graphs were generated by one of three possible algorithms:

- Preferential Attachment (`'PA'`)
- Small World with low probability of rewiring (`'SW_L'`)
- Small World with high probability of rewiring (`'SW_H'`)

Anaylze each of the 5 graphs and determine which of the three algorithms generated the graph.

The `graph_identification` function should return a list of length 5 where each element in the list is either `'PA'`, `'SW_L'`, or `'SW_H'`.

In [4]:
```
def graph_identification():
    methods = []
    for G in P1_Graphs:
        degrees = G.degree()
        degree_values = sorted(set(degrees.values()))
        degree_hist = [list(degrees.values()).count(i) / float(nx.number_of_
        clustering = nx.average_clustering(G)
        shortest_path = nx.average_shortest_path_length(G)

        if len(degree_hist)>10:
            methods.append('PA')
        elif clustering < 0.1:
            methods.append('SW_H')
```