Breakout Activity Week 11: Graphs

In this week's async, we talked about measures on graphs. Today we'll extract data from neo4j and analyze it using Python.

To begin, install 2 python modules: py2neo for connecting to neo4j and networkx for analyze the graph data. Do this by running:

```
pip install py2neo networkx
```

With Neo4J

If you have installed neo4j and loaded the graph data for Lab 11, follow these steps to extract the data from neo4j:

1. Make sure neo4j is started at http://hostname:7474/db/data
2. Open the python interpreter
3. Run the following code

```
import py2neo as pn
import networkx as nx

ngraph = nx.Graph()

def add_to_nx_graph(rec):

ngraph.add_edge(rec.r.start_node.properties["name"],rec.r.en
d_node.properties["name"], \
    weight=rec.r.properties["w"])

pgraph = pn.graph("http:// localhost:7474/db/data")

cypher_query = """MATCH (n)-[r:APPEARED]-(m) RETURN r ORDER
BY r.w DESC SKIP {o} LIMIT {l}"""

for i in range(0,pgraph.size,1000):
  records = pgraph.cypher.execute(cypher_query, o=i,l=1000)
  for r in records:
    add_to_nx_graph(r)

tail = pgraph.size - i
final_records = pgraph.cypher.execute(cypher_query,
o=i,l=tail)
for r in final_records:
  add_to_nx_graph(r)
```

Without Neo4J

If you do not have neo4j storing the Marvel Character data, you can download and import a copy into the python environment. Follow these steps

1. Download the data using:
   ```
   wget https://s3.amazonaws.com/ucbw205data/marvel_characters.gml
   ```
2. Start the python interpreter
3. Run the following code
   ```
   import networkx as nx

   ngraph = nx.read_gml("marvel_characters.gml ")
   ```

Exploring the Graph

We'll use the algorithms in NetworkX to better understand the structure of the graph. A reference to all algorithms can be found here: http://networkx.readthedocs.org/en/latest/reference/algorithms.html

Answer the following questions:

1. What is the size and diameter of the graph?
2. How many connected components are in the graph?
   a. How big is the giant component?
   b. How big is the smallest component?
3. Isolate the giant component as a new graph (Hint: networkx.connected_components returns it set of nodes in the component)
4. Calculate the pagerank for giant component
5. Find the number of communities in the giant component using k-clique.

If you have time: write the Component ID, PageRank, and Community ID of each node back to Neo4J using py2neo.