Project 3: **Social Network Analysis**
Intro to Web Science
50 points

Write a Python script using the networkx library that performs some analysis of Twitter data. You may work in pairs using pair programming or work individually.

Twitter Data

The Twitter data comes from SNAP, the Stanford Large Network Dataset Collection. The data set was originally collected by J. McAuley and J. Leskovec in Learning to Discover Social Circles in Ego Networks. NIPS, 2012.

Download the twitter.tar.gz file from SNAP or from my website and unzip the file:
https://sites.harding.edu/fmccown/classes/comp4750-f19/twitter.zip

The file contains multiple sets of four related files using a unique number at the front. Example: 12831.circle, 12831.edges, 12831.egofeat, 12831.featnames.

The **.edges** file is an edgelist of Twitter of anonymized users. In the following example, user 398874773 "follows" 652193 in Twitter:

```
398874773 652193
18498878 14749606
14305022 8479062
22253 12741
…
```

The **.featnames** file contains a list of hashtags and usernames found in the tweets issued by Twitter users. Example:

```
…
19 #BlackoutSOPA!
20 #CHIbestpaper
21 #DLD12
22 #EC2
23 #FF
```

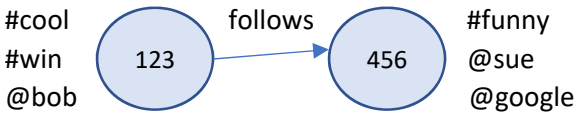
The **.feat** file contains a list of zeroes and ones indicating which features (hashtags or usernames) were tweeted by the Twitter user. The first string is the Twitter node name. In the following example, the user 180505807 used the 3rd and last feature in .featname, and user 369246180 used only the 2nd feature:

```
180505807 0 0 1 0 0 0 0 0 0 … 1
369246180 0 1 0 0 0 0 0 0 0 … 0
14202711 0 0 0 0 0 0 0 0 0 … 0
761 0 0 0 0 0 0 0 0 0 0 1 1 … 0
…
```

Note that this file may contain some users that do not appear in the .edges file.

Build a Graph

Build a directed graph from the Twitter data files where the node contains attributes that match the node's given features. In other words, a node would have attributes "#cool" and "@WhiteHouse" if the associated user tweeted those two strings. Example showing two nodes and their attributes:



The read_edgelist() function can create a directed graph from the edgelist file, but you will need to do some research on how to read from a text file and parse the data so you can add the appropriate properties to each node.

Assume your Python script and data files are in the same directory.

Script Output

Your script should output the following information:

1. Number of nodes, edges, and components (ignoring edge direction)
2. Top 5 users with the most followers (most incoming edges)
3. Top 5 users with betweenness centrality and associated values
4. Top 5 largest cliques and total nodes in the clique (ignoring edges direction)
5. Top 5 longest of the shortest paths between two users who both tweeted the same keyword (hashtag or username)

For each "Top 5", always output the first 5 largest values discovered by the sorted() function, and don't worry about ties.

Look for networkx functions to help you perform the operations above. Example: The networkx function betweenness_centrality() can calculate the betweenness centrality of all nodes. For performing step 5 above, some useful networkx functions are: get_node_attributes(), has_path(), shortest_path_length().

For step 5 above, you first need to get the list of nodes that have the same attributes (list of users who tweeted the same keyword). Then loop through the permutations of the nodes, two at a time, to see if a path exists between any two nodes. Example: If nodes A, C, and E all tweeted the same keyword, then you'd need to see if a path exists between A → C, C → A, A → E, E → A, C → E, E → C. If a path exists, then store the shortest distance between the two nodes. Perhaps the simplest way to do this is to use a dictionary to store key = "node1 node2 keyword" and value = shortest distance. Finally, sort the dictionary on the shortest distance to find the top 5 values.

Make your output match the example formatting below so I can easily compare your script's output with mine. The command line argument is the prefix of the Twitter files your script should use to build

the graph.

```
$ twitter_network.py 12831

Nodes: 244
Edges: 2478
Components: 10

Top 5 most followers:
1. 180505807 - 52
2. 14231571 - 46
3. 1260231 - 46
4. 380 - 44
5. 11178592 - 42

Top 5 betweenness centrality:
1. 174958347 - 4032.3974007790075
2. 1186 - 3587.1066065702034
3. 1678471 - 3189.6854423023096
4. 15583257 - 2954.9008982173764
5. 883301 - 2811.000500360865

Top 5 largest cliques:
1. ['1186', '380', '652193', '10476462', '14202711', '8479062', '6331462', '14367669',
'5854882', '179339999'] - 10
2. ['1186', '380', '652193', '586', '14202711', '8479062', '179339999', '14367669', '5854882',
'6331462'] - 10
3. ['9283582', '15384741', '17729005', '17384098', '14819149', '6253882', '9767472',
'19101100', '15304923'] - 9
4. ['728163', '14172562', '51123', '22253', '2735631', '528', '6735', '8067082', '6503412'] -
9
5. ['728163', '14172562', '51123', '19223', '14178728', '528', '6735', '2735631', '8067082'] -
9

Top 5 longest paths with same keyword:
1. 17561826 - 15543818 - #OWS - 7
2. 15725851 - 15543818 - #kony2012 - 7
3. 13538092 - 10044992 - #sanfrancisco - 7
4. 18976476 - 7461782 - @mikeyk - 7
5. 18976476 - 10044992 - @mikeyk – 7
```

Submit

Submit your working script to Canvas before it is due. Put your name at the top in comments along with your partner (if you pair programmed).