Homework Session III:

1. Download a network of Twitter data on a topic of your choice (mention the topic in you report) with at least 1000 nodes (using the code on the following page), use the mentions (retweets + replies) to get more links between the nodes. Make sure that that there are a decent number of links (connected component with more than 500 nodes), and the clustering coefficient is larger than 0.

2. Use SNAP to calculate
    a. The most central nodes (with at least three different methods)
    b. The average path length, diameter and effective diameter
    c. The innermost k-shell (size and value)
    d. The clustering coefficient

3. Report metrics of point 2 and the degree distribution (plus Power-law fit or equivalent, use http://tuvalu.santafe.edu/~aaronc/powerlaws/ similar to the example used in class and the code from http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0085777

4. Compare the network with similar networks in size generated with
    a. Preferential Attachment
    b. Configuration model
    c. Node Rewiring
    d. Erdos-Renyi random graph

5. Interpret the results

---------- code to download tweets ----------

needs access tokens, check: http://www.zdidit.com/how-to-create-twitter-app/

and https://apps.twitter.com/

```
import sys
import tweepy
import json
consumer_key = 'XXX'
consumer_secret = 'XXX'
access_token = 'XXX'
access_token_secret = 'XXX'
class StdOutListener(tweepy.StreamListener):
    def on_status(self, status):
        try:
             #tweet_json = json.loads(json.dumps(status._json))
             #print tweet_json
             print json.dumps(status._json)
        except:
             print "Unexpected error:", sys.exc_info()[0]
        return True
    def on_error(self, status_code):
        print('Got an error with status code: ' + str(status_code))
        return False # To continue listening
    def on_timeout(self):
        print('Timeout...')
        return False # To continue listening
if __name__ == '__main__':
    listener = StdOutListener()
    auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
    auth.set_access_token(access_token, access_token_secret)
    stream = tweepy.Stream(auth, listener)
    #stream.filter(locations=[-6.38,49.87,1.77,55.81])
    # coordinates of barcelona
    stream.filter(track=['hello'])
```