

# testByJupyter

April 11, 2018

```
In [2]: # ----- Now your turn -----#
# Let's us say you are U19886 and you want to work at Google.
# Explore, implement your strategy to be helped in getting this job at Google.
import networkx as nx
import matplotlib.pyplot as plt
from matplotlib import pylab
import numpy as np
import pickle

#----- Let's take a quick NetworkX tour -----#
# load the graph
G = nx.read_gexf("mediumLinkedin.gexf")

# load the profiles. 3 files for each type of attribute
# Some nodes in G have no attributes, or just location, or all of same
# Some nodes may have 2 or more colleges or employers, so we
# use dictionaries to store the attributes
college = {}
location = {}
employer = {}
# The dictionaries are loaded as dictionaries from the disk (see pickle in Python doc)
with open('mediumCollege.pickle', 'rb') as handle:
    college = pickle.load(handle)
with open('mediumLocation.pickle', 'rb') as handle:
    location = pickle.load(handle)
with open('mediumEmployer.pickle', 'rb') as handle:
    employer = pickle.load(handle)

print("Nb of users with one or more attribute college: %d" % len(college))
print("Nb of users with one or more attribute location: %d" % len(location))
print("Nb of users with one or more attribute employer: %d" % len(employer))

def findUserInfo(employeeNum):
    print('For User = ' + employeeNum)
    if employeeNum in [k for k in college.keys()]:
        print('colleges exists')
        print(college[employeeNum])
```

```

if employeeNum in [k for k in employer.keys()]:
    print('employer exists')
    print(employer[employeeNum])
if employeeNum in [k for k in location.keys()]:
    print('location exists')
    print(location[employeeNum])
print('\n')

```

Nb of users with one or more attribute college: 540  
 Nb of users with one or more attribute location: 811  
 Nb of users with one or more attribute employer: 730

```

In [3]: clientNum = 'U19886'
        findUserInfo(clientNum)

```

```

For User = U19886
location exists
['rockford illinois area']

```

```

In [4]: # string matching Functions
def strMatchDict(str1,dict2 , jobNow = False ):
    users = []
    if dict2 != None :
        if jobNow == True:
            for key in dict2.keys():
                if dict2[key][0].find(str1) >= 0:
                    users.append(key)
        else:
            for key in dict2.keys():
                for str2 in dict2[key]:
                    if str2.find(str1) >= 0:
                        users.append(key)
    return users

def strMatchList(str1,list2 ):
    for i in range(0,len(list2)):
        if list2[i].find(str1)>=0 and i==0:
            return 'is working in' + ' ' +str1
        elif list2[i].find(str1)>=0 and i != 0:
            return 'was working in' + ' ' +str1
    return 'not in ' + str1

```

```

In [5]: ggEmAll = strMatchDict('google',employer)
        print('all google workers = ' + str(len(ggEmAll)))
        ggEmNow = strMatchDict('google',employer,True)
        print('actual google workers = ' +str(len(ggEmNow)))

```

```
all google workers = 35
actual google workers = 23
```

```
In [6]: # Test Finished
        # Implementation of BFS
        import queue as Q

        endPoint = []
        dicPaths = {}
        oldNum = []
        q = Q.Queue()
        q.put(clientNum)

        ggEmDegree = []
        for n in ggEmNow:
            ggEmDegree.append((G.degree(n), n ))

        ggEmDegree = [m for (n,m) in sorted(ggEmDegree,reverse=True)]

        for i in range(0,3000):
            user = q.get()
            oldNum.append(user)
            ngbs = G.neighbors(user)
            for n in ngbs:
                if n not in oldNum:
                    q.put(n)
                    dicPaths[n] = user
                    if n in ggEmDegree[:10] and n not in endPoint:
                        print(n+' worked in Google')
                        endPoint.append(n)
```

```
U7202 worked in Google
U24154 worked in Google
U19913 worked in Google
U24080 worked in Google
U27661 worked in Google
U27494 worked in Google
U4568 worked in Google
U3955 worked in Google
U11558 worked in Google
U4010 worked in Google
```

```
In [8]: # Get and find the path!
        Roads = {}
        for i in range(0,len(endPoint)):
            Roads[endPoint[i]] = [endPoint[i]]
```

```

beforeP = dicPaths[endPoint[i]]
while(beforeP != clientNum):
    Roads[endPoint[i]].insert(0,beforeP)
    beforeP = dicPaths[beforeP]
Roads[endPoint[i]].insert(0,clientNum)

```

```

In [32]: # Fastest Path and EndPointDegree
Way1 = [(len(Roads[key]),key) for key in Roads.keys()]
Way1 = sorted(Way1)
print(Way1[0][1] + '\s Degree is ' + str(G.degree(Way1[0][1])))
print('\n')
print('Fastest Path is \n\t' + str(Roads[Way1[0][1]]))

```

U7202's Degree is 5

Fastest Path is  
['U19886', 'U7091', 'U7167', 'U7024', 'U7202']

```

In [16]: # Show fastest path's infomation
for i in range(0,len(Roads[Way1[0][1]])):
    findUserInfo(Roads[Way1[0][1]][i])

```

For User = U19886  
location exists  
['rockford illinois area']

For User = U7091  
employer exists  
['illinois college advising corps']  
location exists  
['urbana-champaign illinois area']

For User = U7167  
colleges exists  
['university of illinois']  
employer exists  
['champaign-urbana special recreation', 'old navy']  
location exists  
['urbana-champaign illinois area']

For User = U7024  
colleges exists

```

['university of illinois at urbana-champaign']
employer exists
['university of illinois at urbana-champaign']
location exists
['urbana-champaign illinois area']

```

```

For User = U7202
employer exists
['google', 'university of illinois at urbana-champaign']
location exists
['san francisco bay area']

```

```

In [31]: # Biggest Degree Path
Way2 = [(G.degree(key),key) for key in Roads.keys()]
Way2 = sorted(Way2,reverse =True)
print(Way2[0][1] + '\s Degree is ' + str(G.degree(Way2[0][1])))
print('\n')
print('Biggest degree Path is \n\t' + str(Roads[Way2[0][1]]))

```

```

U3955's Degree is 19

```

```

Biggest degree Path is
['U19886', 'U7091', 'U7167', 'U7024', 'U7319', 'U7256', 'U27287',
 'U27585', 'U27661', 'U11597', 'U3919', 'U3955']

```

```

In [30]: # Show Biggest degree path's infomation
for i in range(0,len(Roads[Way2[0][1]])):
    findUserInfo(Roads[Way2[0][1]][i])

```

```

For User = U19886
location exists
['rockford illinois area']

```

```

For User = U7091
employer exists
['illinois college advising corps']
location exists
['urbana-champaign illinois area']

```

```

For User = U7167
colleges exists

```

```
['university of illinois']
employer exists
['champaign-urbana special recreation', 'old navy']
location exists
['urbana-champaign illinois area']
```

```
For User = U7024
colleges exists
['university of illinois at urbana-champaign']
employer exists
['university of illinois at urbana-champaign']
location exists
['urbana-champaign illinois area']
```

```
For User = U7319
colleges exists
['university of illinois at urbana-champaign']
employer exists
['amazon', 'neustar inc.', 'university of illinois at urbana-champaign']
location exists
['urbana-champaign illinois area']
```

```
For User = U7256
colleges exists
['university of illinois at urbana-champaign']
employer exists
['university of illinois at urbana-champaign']
location exists
['urbana-champaign illinois area']
```

```
For User = U27287
colleges exists
['university of illinois at urbana-champaign', 'shanghai jiao tong university',
 'university of new mexico', 'hong kong university of science and technology']
employer exists
['university of illinois at urbana-champaign']
location exists
['urbana-champaign illinois area']
```

```
For User = U27585
colleges exists
['east china normal university software engineering institute']
employer exists
```

```
['apple', 'university of pittsburgh school of information science', 'haoban digital co.']  
location exists  
['san francisco bay area']
```

```
For User = U27661  
colleges exists  
['shanghai jiao tong university']  
employer exists  
['google', 'at&t labs inc.', 'university of michigan']  
location exists  
['san francisco bay area']
```

```
For User = U11597  
employer exists  
['baidu inc.']  
location exists  
['beijing city china']
```

```
For User = U3919  
colleges exists  
['fudan university']  
employer exists  
['mediav', 'yahoo! beijing r&d', 'alibaba cloud computing company',  
    'microsoft global technology supporting center']  
location exists  
['shanghai city china']
```

```
For User = U3955  
employer exists  
['google']  
location exists  
['china']
```