import networkx as nx

import matplotlib.pyplot as plt

def assign(g):

for each in g.nodes():

g.node[each]['action']='B'

def color(g):

colors=[]

for each in g.nodes():

if g.node[each]['action']=='B':

colors.append('green')

else:

colors.append('red')

return colors

def count(g):

c=0

for each in g.nodes():

if g.node[each]['action']=='A':

c+=1

return c

def intro(g,r):

for each in r:

g.node[each]['action']='A'

def replace(g):

#Taking the payoff for A is 4

#Taking the payoff for B is 2

a=9

b=5

for each in g.nodes():

sum1=0

sum2=0

n=[x for x in g.neighbors(each)]

for var in n:

if g.node[var]['action']=='A':

sum1=sum1+a

else:

sum2=sum2+b

if sum2<sum1:

g.node[each]['action']='A'

else:

g.node[each]['action']='B'

def disp(g):

col=color(g)

nx.draw(g,with\_labels=1,node\_color=col,node\_size=800)

plt.show()

g=nx.read\_gml("tester\_graph.gml")

d=list(g.degree())

d.sort(key=lambda x:x[1],reverse=True)

assign(g)

disp(g)

#code for selecting the key people to complete the cascade

'''for u in g.nodes():

for v in g.nodes():

if (u<v):

random\_take=[]

random\_take.append(u)

random\_take.append(v)

intro(g,random\_take)

print random\_take

while(1):

ptr+=1

replace(g)

c=count(g)

if c==0:

print 'Terminated as Idea A has been completely dismissed'

break

elif c==len(g):

print 'Terminated as Idea A has been completely adopted'

break

del random\_take'''

random\_take=['1','8']

intro(g,random\_take)

disp(g)

ptr=0

while(1):

ptr+=1

replace(g)

disp(g)

c=count(g)

if c==0:

print 'Terminated as Idea A has been completely dismissed'

break

elif c==len(g):

print 'Terminated as Idea A has been completely adopted'

break

print 'Action over after ',ptr,' iterations'

disp(g)