import networkx as nx

import matplotlib.pyplot as plt

import random as r

def display(g,i,ne,string):

if (i=='' and ne==''):

prev\_node=g.nodes()

new\_node=[]

prev\_edge=g.edges()

new\_edge=[]

else:

new\_node=[i]

new\_edge=ne

prev\_edge=list(set(g.edges())-set(new\_edge)-set([(b,a) for (a,b) in new\_edge]))

prev\_node=list(set(g.nodes())-set(new\_node))

pos=nx.circular\_layout(g)

nx.draw\_networkx\_nodes(g,pos,nodelist=prev\_node,node\_color='r')

nx.draw\_networkx\_nodes(g,pos,nodelist=new\_node,node\_color='g')

nx.draw\_networkx\_edges(g,pos,edgelist=prev\_edge,edge\_color='r')

nx.draw\_networkx\_edges(g,pos,edgelist=new\_edge,edge\_color='g',style='dashdot')

plt.savefig(str(string))

plt.close()

def node\_barabasi(g,n,m0):

m=m0-1

ref=0

for var in range(m0+1,n+1):

ref+=1

g.add\_node(var)

deg=nx.degree(g)

prob={}

sum=0

for n,p in deg:

sum+=p

for each in g.nodes():

prob[each]=float(deg[each])/float(sum)

cum\_prob=[]

index=0

for n,f in prob.items():

temp=[n,index]

index+=f

cum\_prob.append(temp)

new\_edge=[]

no\_edge=0

target\_node=[]

k=0

while(no\_edge<m):

no=r.random()

prev\_cum=0

k=0

while(not(no>prev\_cum and no<=cum\_prob[k][1])):

prev\_cum=cum\_prob[k][1]

k+=1

node=cum\_prob[k][0]

print node

if node in target\_node:

continue

else:

target\_node.append(node)

g.add\_edge(var,node)

no\_edge+=1

new\_edge.append((var,node))

display(g,var,new\_edge,ref)

print var-m0,' node added'

def main():

'''n=input('Enter the value of n')'''

n=20

m0=r.randint(3,n/5)

g=nx.path\_graph(m0)

display(g,'', '',0)

node\_barabasi(g,n,m0)

main()