

```

"""
@author: campioni

Cap 6

"""

import sys
import networkx as nx
import numpy as np
sys.path.insert(0, ' D/PYTHON/Programmi/code')
from networkx.algorithms import approximation as app
from thinkstats2 import Pmf
import thinkplot
import matplotlib.pyplot as plt

nome='D:/PYTHON/facebook_combined.txt.gz' # carica FB e lo assegna a nome

def degrees(G): # calcola i gradi del grafo
    return[G.degree(u) for u in G]
def read_graph(filename): # funzione di lettura e grafo del file
    G=nx.Graph()
    array=np.loadtxt(filename, dtype=int)
    G.add_edges_from(array)
    return G

fb = read_graph(nome) # assegna a fb il grafo
Pmf_fb=Pmf(degrees(fb)) # calcola PMF per FB

ws=nx.watts_strogatz_graph(4039,44,0.05)
Pmf_ws=Pmf(degrees(ws)) # calcola PMF per WS

print()

print('-----')

print()
print('I grafici sono i seguenti')
print()
print('-----')

print()

fig, ax = plt.subplots(facecolor='teal', alpha=0.1)
plt.style.use('ggplot') #sottofondo grigio e griglia bianca
plt.tick_params(labelcolor='y')
plt.xlabel('Gradi', color='y')
thinkplot.Pdf(Pmf_ws, label='Grafico WS - Gaussiana', color='r')
thinkplot.Show()

fig, ax = plt.subplots(facecolor='teal', alpha=0.1)
plt.style.use('ggplot') #sottofondo grigio e griglia bianca
plt.tick_params(labelcolor='y')
plt.xlabel('Gradi', color='y')
thinkplot.Pdf(Pmf_fb, label='Facebook - Legge di potenza', color='r')
thinkplot.Show()

```