

```

"""
Created on Wed Mar  4 18:27:15 2020
Cap 11
@author: Massimo Campioni
"""

import networkx as nx
import numpy as np

G=nx.Graph()
print()

N=int(input('numero N di geni = '))
k=int(input('numero k (<N) di geni collegati = '))
print()
f=k+1

print()

def ada(N):
    w=0
    for h in range(N):
        w=w+np.random.rand()/(N)
    return w

def LISTA_ADA(N):
    A=[]
    for i in range (2**f):
        a=round(ada(N),5)
        A.append(a)
    return A

A=LISTA_ADA(N)

G=nx.hypercube_graph(f)          #genera un ipercubo
nx.draw(G,node_color='red',
        node_size=2000,
        with_labels=True)

P=[]

Z=[]
Z1=[]
Z2=[]
valori=[]
picco=[]
nodo=[]

b=list(G)
c=0
if k==0:
    print('Unico picco di valore ', '%5f' % np.max(A), 'al nodo n° ',
A.index(np.max(A)))
else:
    for i in b:
        P.append([n for n in G.neighbors(i)]) #crea una lista dei k vicini
        #degli 2**k nodi
    for i in range (2**f): # converte la tupla interna a P in una lista di
numeri
        for h in range(f): # di k+1 x 2**(k+1) valori
            Z=("".join(str(h) for h in P[i][h]))
            Z1.append(Z)

    for i in range(2**f*f):          #conversione di Z1 da binario a decimale

```

```

Z2.append(int(Z1[i],2))

for i in Z2:      #e identificati da P
    valori.append(A[i])

for h in range(2**f):
    if A[h]>=np.max(valori[f*h:f*h+f]):
        picco.append(valori[f*h])
        nodo.append(h)
        c+=1
    ...

=====
Si stampano i risultati
=====
...

print('I picchi relativi sono: ' , c ,', specificatamente:')
print()
for i in range(c):
    print('Nodo ',Z1[i], ' --- valore   ', '%5f' % picco[i])

print()
print('=====')
print()

print('Il picco massimo è ', '%5f' % np.max(picco))

```