EulerCharacteristicAlgo

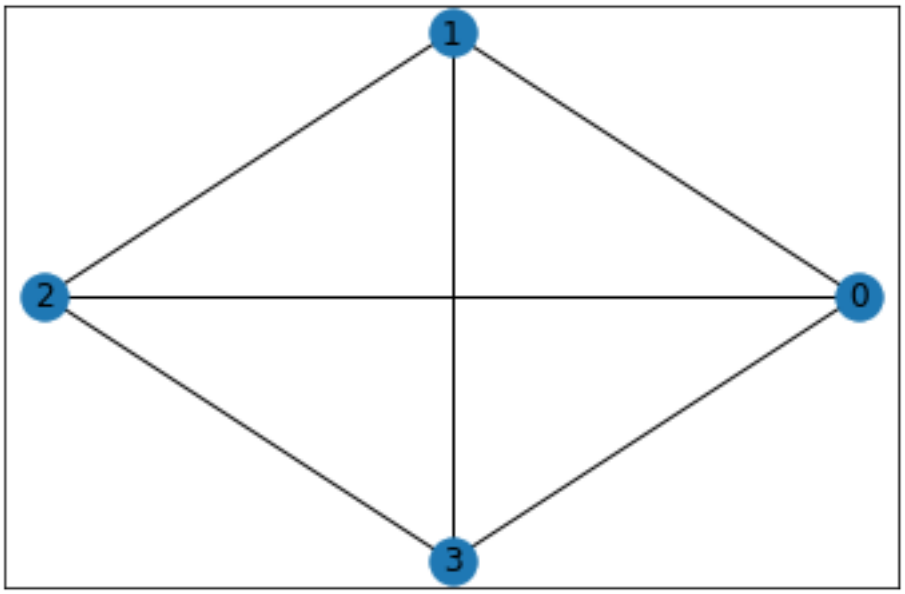
April 4, 2022

[66]:
```
from itertools import combinations
import networkx as nx
import numpy as np
import matplotlib.pyplot as plt


#Finding number of k-cliques in a Graph
#Found this code online at the following link: https://iq.opengenus.org/
 ↪algorithm-to-find-cliques-of-a-given-size-k/
def k_cliques(graph):
    # 2-cliques
    cliques = [{i, j} for i, j in graph.edges() if i != j]
    k = 2
    while cliques:
        yield k, cliques

        # merge k-cliques into (k+1)-cliques
        cliques_1 = set()
        for u, v in combinations(cliques, 2):
            w = u ^ v
            if len(w) == 2 and graph.has_edge(*w):
                cliques_1.add(tuple(u | w))
        # remove duplicates
        cliques = list(map(set, cliques_1))
        k += 1
def print_cliques(graph, size_k):
    for k, cliques in k_cliques(graph):
        if k == size_k:
            return len(cliques)

#Function for computing the Euler Characteristic for a Graph G
def Euler_Char(G):
    """Computes Euler Characteristic for graph G"""
    chi = 0
    chi = chi + G.number_of_nodes()
    #print(print_cliques(G, 3))
    for i in range(2, G.number_of_nodes()+1):
```

```
        if type(print_cliques(G, i)) == type(1):
            chi = chi + (-1)**(i+1) * print_cliques(G, i)
        else:
            break
    return chi
```

[67]:
```
#Generates Erdos Reyni Graph and computes its Euler Characteristic.
G = nx.erdos_renyi_graph(4, .9, seed=None, directed=False)

print(f"Euler Characteristic of G: {Euler_Char(G)}")
nx.draw_networkx(G, pos=nx.circular_layout(G))
```

```
Euler Characteristic of G: 1
```



[ ]:
```
#Returns possible p for which there is a topological phase transition (Euler␣
 ↪entropy becomes singular)
k = 10
S = np.linspace(0, 1, 100)

for j in S:
    X = 0
    for i in range(k):
        G = nx.erdos_renyi_graph(15, j, seed=None, directed=False)
        X = X + Euler_Char(G)
    avg = X/k
    if np.abs(avg) <= .1:
        print(f"Possible phase transition at p = {j} with average :{avg}")
```

```
print('a')
```

```
Possible phase transition at p = 0.16161616161616163 with average :0.0
```