

In [1]:

```
%matplotlib inline
import networkx as nx
import matplotlib.cm as cm
import matplotlib.pyplot as plt
```

In [2]:

```
import networkx as nx
```

```
G=nx.Graph()
G.add_node("spam")
G.add_edge(1,2)
```

```
print(G.nodes())
```

```
print(G.edges())
```

```
['spam', 1, 2]
[(1, 2)]
```

In [3]:

```
nx.draw(G, with_labels = True)
```

C:\Users\se\Anaconda3\lib\site-packages\networkx\drawing\nx_pylab.py:126: Matplotlib DeprecationWarning: pyplot.hold is deprecated.

Future behavior will be consistent with the long-time default:
plot commands add elements without first clearing the
Axes and/or Figure.

```
b = plt.ishold()
```

C:\Users\se\Anaconda3\lib\site-packages\networkx\drawing\nx_pylab.py:138: Matplotlib DeprecationWarning: pyplot.hold is deprecated.

Future behavior will be consistent with the long-time default:
plot commands add elements without first clearing the
Axes and/or Figure.

```
plt.hold(b)
```

C:\Users\se\Anaconda3\lib\site-packages\matplotlib__init__.py:917: UserWarning: axes.hold is deprecated. Please remove it from your matplotlibrc and/or style files.

```
warnings.warn(self.msg_depr_set % key)
```

C:\Users\se\Anaconda3\lib\site-packages\matplotlib\rcsetup.py:152: UserWarning: axes.hold is deprecated, will be removed in 3.0

```
warnings.warn("axes.hold is deprecated, will be removed in 3.0")
```




In [4]:

```
G = nx.Graph()
n = 0
with open ('/Users/se/.ipython/www.dat.gz.txt') as f:
    for line in f:
        n += 1
        #if n % 10**4 == 0:
            #flushPrint(n)
        x, y = line.rstrip().split(' ')
        G.add_edge(x, y)
```

In [5]:

```
nx.info(G)
```

Out[5]:

```
'Name: \nType: Graph\nNumber of nodes: 325729\nNumber of edges: 1117563\nAverage degree: 6.8619'
```

In [6]:

```
G = nx.karate_club_graph()

clubs = [G.node[i]['club'] for i in G.nodes()]
colors = []
for j in clubs:
    if j == 'Mr. Hi':
        colors.append('r')
    else:
        colors.append('g')

nx.draw(G, with_labels = True, node_color = colors)
```

C:\Users\se\Anaconda3\lib\site-packages\networkx\drawing\nx_pylab.py:126: Matplotlib DeprecationWarning: pyplot.hold is deprecated.

Future behavior will be consistent with the long-time default:
plot commands add elements without first clearing the
Axes and/or Figure.

```
b = plt.ishold()
```

C:\Users\se\Anaconda3\lib\site-packages\networkx\drawing\nx_pylab.py:138: Matplotlib DeprecationWarning: pyplot.hold is deprecated.

Future behavior will be consistent with the long-time default:
plot commands add elements without first clearing the
Axes and/or Figure.

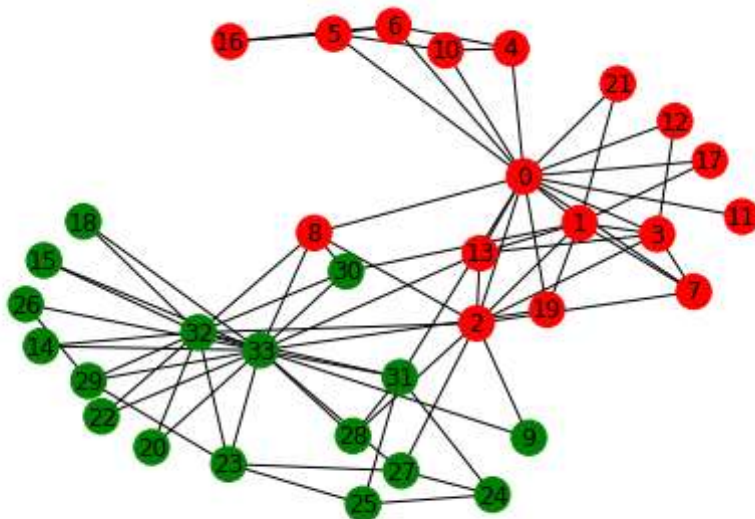
```
plt.hold(b)
```

C:\Users\se\Anaconda3\lib\site-packages\matplotlib__init__.py:917: UserWarning: axes.hold is deprecated. Please remove it from your matplotlibrc and/or style files.

```
warnings.warn(self.msg_depr_set % key)
```

C:\Users\se\Anaconda3\lib\site-packages\matplotlib\rcsetup.py:152: UserWarning: axes.hold is deprecated, will be removed in 3.0

```
warnings.warn("axes.hold is deprecated, will be removed in 3.0")
```



In [7]:

```
G.node[1]
```

Out[7]:

```
{'club': 'Mr. Hi'}
```

In [8]:

```
list(G.edge.keys())[:3]
```

Out[8]:

```
[0, 1, 2]
```

In [9]:

```
nx.info(G)
```

Out[9]:

```
"Name: Zachary's Karate Club\nType: Graph\nNumber of nodes: 34\nNumber of edges: 78\nAverage degree: 4.5882"
```

In [10]:

```
G.nodes()[:10]
```

Out[10]:

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [11]:

```
G.edges()[:3]
```

Out[11]:

```
[(0, 1), (0, 2), (0, 3)]
```

In [12]:

```
G.neighbors(1)
```

Out[12]:

```
[0, 2, 3, 7, 13, 17, 19, 21, 30]
```

In [13]:

```
nx.average_shortest_path_length(G)
```

Out[13]:

```
2.408199643493761
```

In [14]:

```
nx.diameter(G)
```

Out[14]:

5

In [15]:

```
nx.density(G)
```

Out[15]:

0.13903743315508021

In [16]:

```
nodeNum = len(G.nodes())  
edgeNum = len(G.edges())  
  
2.0*edgeNum/(nodeNum * (nodeNum - 1))
```

Out[16]:

0.13903743315508021

In [17]:

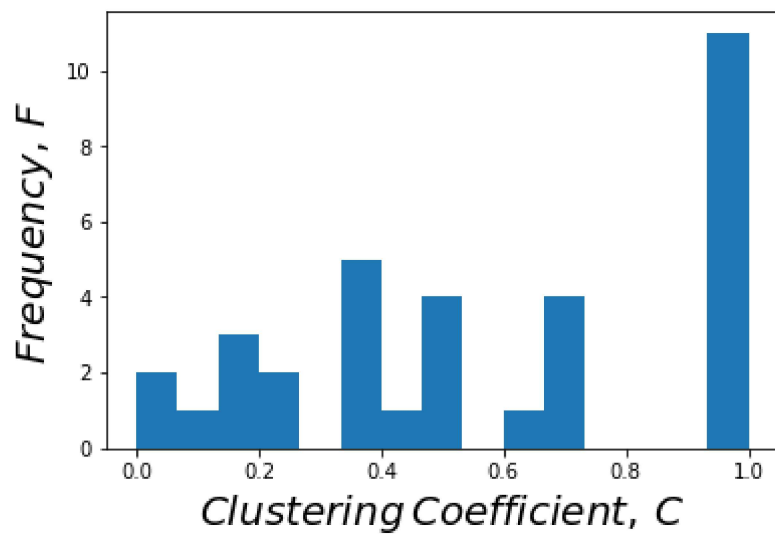
```
cc = nx.clustering(G)  
list(cc.items())[:5]
```

Out[17]:

```
[(0, 0.15),  
 (1, 0.3333333333333333),  
 (2, 0.24444444444444444),  
 (3, 0.6666666666666666),  
 (4, 0.6666666666666666)]
```

In [18]:

```
plt.hist(list(cc.values()), bins = 15)
plt.xlabel('$Clustering \ Coefficient, \ C$', fontsize = 20)
plt.ylabel('$Frequency, \ F$', fontsize = 20)
plt.show()
```



In [19]:

```
nx.degree_assortativity_coefficient(G)
```

Out[19]:

-0.47561309768461457

In [20]:

```
Ge=nx.Graph()
Ge.add_nodes_from([0, 1], size=2)
Ge.add_nodes_from([2, 3], size=3)
Ge.add_edges_from([(0, 1), (2, 3)])
print(nx.numeric_assortativity_coefficient(Ge, 'size'))
```

1.0

In [21]:

```
from collections import defaultdict
import numpy as np

l=defaultdict(list)
g = nx.karate_club_graph()

for i in g.nodes():
    k = []
    for j in g.neighbors(i):
        k.append(g.degree(j))
    l[g.degree(i)].append(np.mean(k))

x = l.keys()
y = [np.mean(i) for i in l.values()]

plt.plot(x, y, 'r-o', label = '$Karate\;Club$')
plt.legend(loc=1, fontsize=10, numpoints=1)
plt.xscale('log'); plt.yscale('log')
plt.ylabel(r'$\langle k \rangle$', fontsize = 20)
plt.xlabel('$k$', fontsize = 20)
plt.show()
```

```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-21-53cc294055eb> in <module>()
     14 y = [np.mean(i) for i in l.values()]
     15
--> 16 plt.plot(x, y, 'r-o', label = '$Karate\;Club$')
     17 plt.legend(loc=1, fontsize=10, numpoints=1)
     18 plt.xscale('log'); plt.yscale('log')

C:\Users\se\Anaconda3\lib\site-packages\matplotlib\pyplot.py in plot(*args, **kwarg
s)
    3316
    3317         mplDeprecation)
    3318     try:
-> 3318         ret = ax.plot(*args, **kwargs)
    3319     finally:
    3320         ax._hold = washold

C:\Users\se\Anaconda3\lib\site-packages\matplotlib\__init__.py in inner(ax, *args, *
**kwargs)
    1889         warnings.warn(msg % (label_namer, func.__name__),
    1890                         RuntimeWarning, stacklevel=2)
-> 1891         return func(ax, *args, **kwargs)
    1892     pre_doc = inner.__doc__
    1893     if pre_doc is None:

C:\Users\se\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py in plot(self, *arg
s, **kwargs)
    1404         kwargs = cbook.normalize_kwargs(kwargs, _alias_map)
    1405
-> 1406         for line in self._get_lines(*args, **kwargs):
    1407             self.add_line(line)
    1408             lines.append(line)

C:\Users\se\Anaconda3\lib\site-packages\matplotlib\axes\_base.py in
grab_next_args(self, *args, **kwargs)
    405         return
    406         if len(remaining) <= 3:
```

```
--> 407         for seg in self._plot_args(remaining, kwargs):
408             yield seg
409         return
```

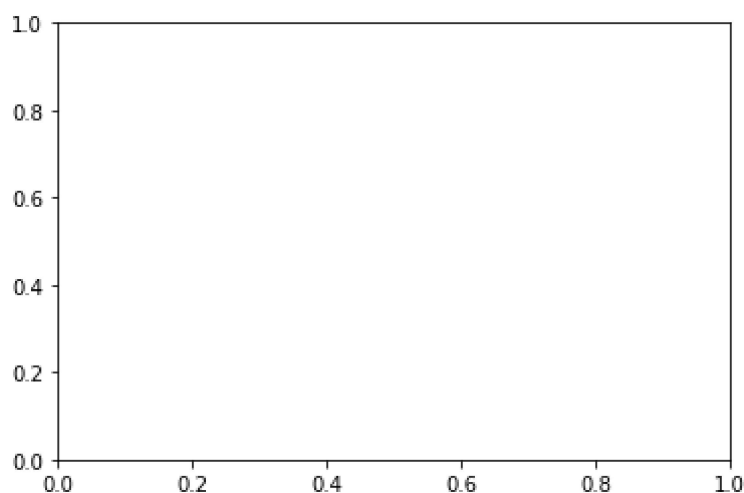
```
C:\Users\se\Anaconda3\lib\site-packages\matplotlib\axes\_base.py in _plot_args(self,
tup, kwargs)
```

```
383         x, y = index_of(tup[-1])
384
--> 385         x, y = self._xy_from_xy(x, y)
386
387         if self.command == 'plot':
```

```
C:\Users\se\Anaconda3\lib\site-packages\matplotlib\axes\_base.py in _xy_from_xy(self, x, y)
```

```
242         if x.shape[0] != y.shape[0]:
243             raise ValueError("x and y must have same first dimension, but "
--> 244                             "have shapes {} and {}".format(x.shape, y.shape))
245
246         if x.ndim > 2 or y.ndim > 2:
            raise ValueError("x and y can be no greater than 2-D, but have "
```

ValueError: x and y must have same first dimension, but have shapes (1,) and (11,)



In [22]:

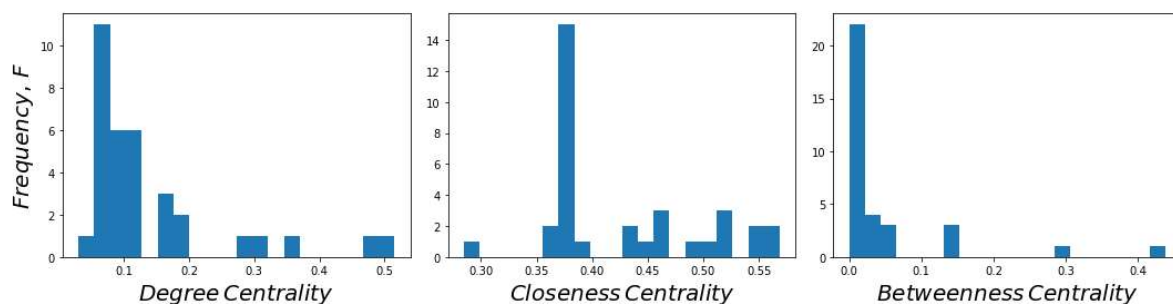
```
dc = nx.degree_centrality(G)
closeness = nx.closeness_centrality(G)
betweenness= nx.betweenness_centrality(G)
```

In [23]:

```
fig = plt.figure(figsize=(15, 4), facecolor='white')
ax = plt.subplot(1, 3, 1)
plt.hist(list(dc.values()), bins = 20)
plt.xlabel('$Degree \, Centrality$', fontsize = 20)
plt.ylabel('$Frequency, \, F$', fontsize = 20)

ax = plt.subplot(1, 3, 2)
plt.hist(list(closeness.values()), bins = 20)
plt.xlabel('$Closeness \, Centrality$', fontsize = 20)

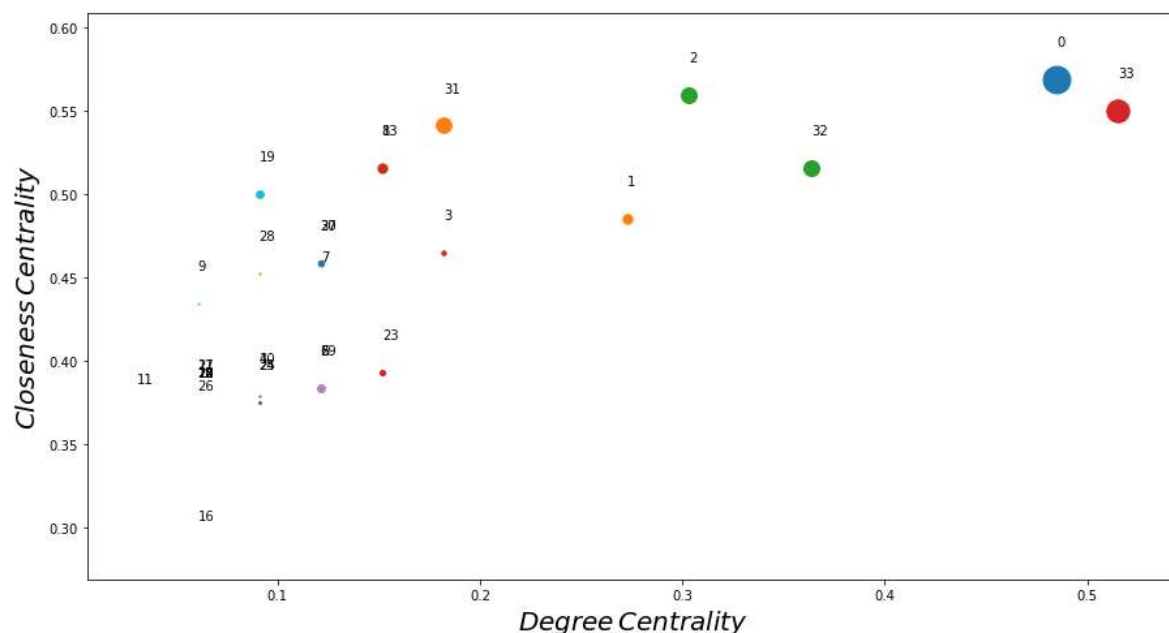
ax = plt.subplot(1, 3, 3)
plt.hist(list(betweenness.values()), bins = 20)
plt.xlabel('$Betweenness \, Centrality$', fontsize = 20)
plt.tight_layout()
plt.show()
```



In [24]:

```
fig = plt.figure(figsize=(15, 8), facecolor='white')

for k in betweenness:
    plt.scatter(dc[k], closeness[k], s = betweenness[k]*1000)
    plt.text(dc[k], closeness[k]+0.02, str(k))
plt.xlabel('$Degree \, Centrality$', fontsize = 20)
plt.ylabel('$Closeness \, Centrality$', fontsize = 20)
plt.show()
```

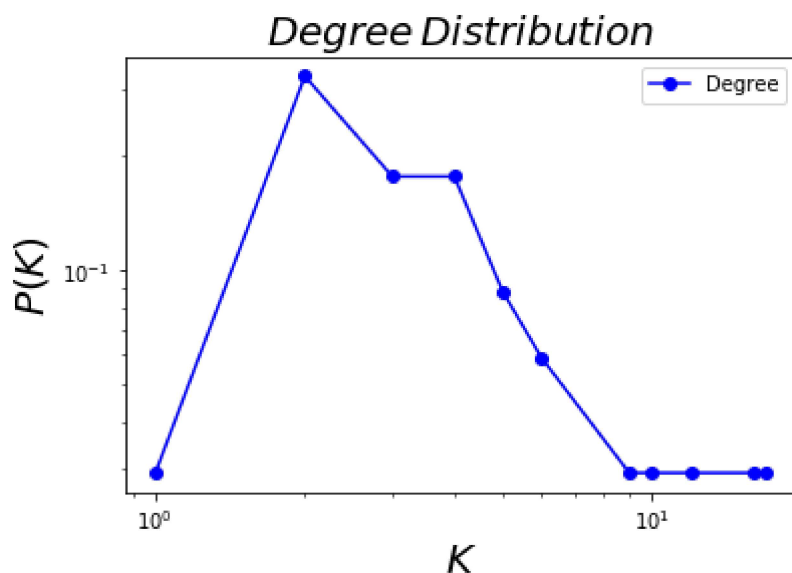


In [25]:

```
from collections import defaultdict
import numpy as np

def plotDegreeDistribution(G):
    degs = defaultdict(int)
    for i in G.degree().values(): degs[i]+=1
    items = sorted ( degs.items () )
    x, y = np.array(items).T
    y_sum = np.sum(y)
    y = [float(i)/y_sum for i in y]
    plt.plot(x, y, 'b-o')
    plt.xscale('log')
    plt.yscale('log')
    plt.legend(['Degree'])
    plt.xlabel(' $K$', fontsize = 20)
    plt.ylabel(' $P(K)$', fontsize = 20)
    plt.title(' $Degree$, Distribution$', fontsize = 20)
    plt.show()
```

```
G = nx.karate_club_graph()
plotDegreeDistribution(G)
```



In [26]:

```
import networkx as nx
import matplotlib.pyplot as plt
RG = nx.random_graphs.random_regular_graph(3, 200)
pos = nx.spectral_layout(RG)
nx.draw(RG, pos, with_labels=False, node_size = 30)
plt.show()
```

C:\Users\se\Anaconda3\lib\site-packages\networkx\drawing\nx_pylab.py:126: Matplotlib

DeprecationWarning: pyplot.hold is deprecated.

Future behavior will be consistent with the long-time default:

plot commands add elements without first clearing the

Axes and/or Figure.

b = plt.ishold()

C:\Users\se\Anaconda3\lib\site-packages\networkx\drawing\nx_pylab.py:138: Matplotlib

DeprecationWarning: pyplot.hold is deprecated.

Future behavior will be consistent with the long-time default:

plot commands add elements without first clearing the

Axes and/or Figure.

plt.hold(b)

C:\Users\se\Anaconda3\lib\site-packages\matplotlib__init__.py:917: UserWarning: axe

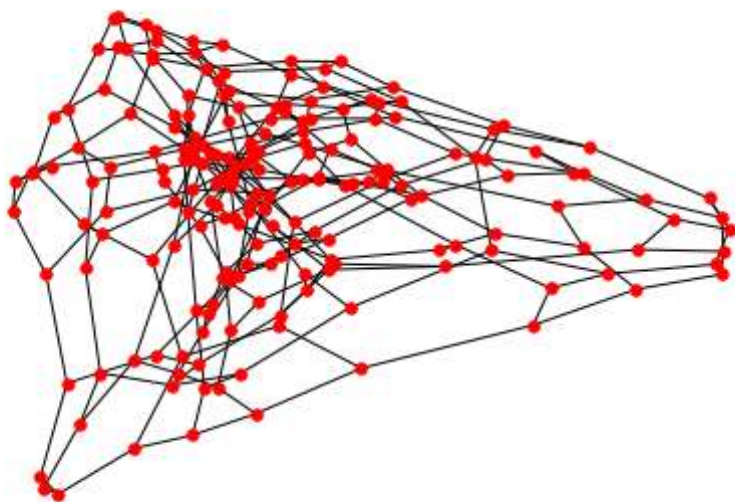
s.hold is deprecated. Please remove it from your matplotlibrc and/or style files.

warnings.warn(self.msg_depr_set % key)

C:\Users\se\Anaconda3\lib\site-packages\matplotlib\rcsetup.py:152: UserWarning: axe

s.hold is deprecated, will be removed in 3.0

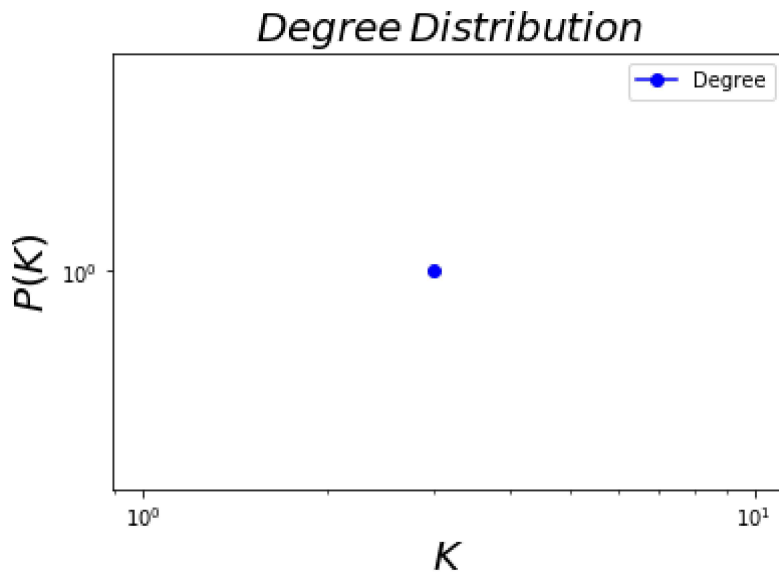
warnings.warn("axes.hold is deprecated, will be removed in 3.0")



In [27]:

```
plotDegreeDistribution(RG)
```

```
C:\Users\se\Anaconda3\lib\site-packages\matplotlib\axes\_base.py:3179: UserWarning:
  Attempting to set identical bottom==top results
  in singular transformations; automatically expanding.
bottom=1.0, top=1.0
'bottom=%s, top=%s') % (bottom, top))
```



In [28]:

```
import networkx as nx
import matplotlib.pyplot as plt
ER = nx.random_graphs.erdos_renyi_graph(200, 0.05)  #生成包含20个节点、以概率0.2连接的随机图
pos = nx.shell_layout(ER)  #定义一个布局, 此处采用了shell布局方式
nx.draw(ER, pos, with_labels=False, node_size = 30)
plt.show()
```

C:\Users\se\Anaconda3\lib\site-packages\networkx\drawing\nx_pylab.py:126: Matplotlib

DeprecationWarning: pyplot.hold is deprecated.

Future behavior will be consistent with the long-time default:

plot commands add elements without first clearing the

Axes and/or Figure.

b = plt.ishold()

C:\Users\se\Anaconda3\lib\site-packages\networkx\drawing\nx_pylab.py:138: Matplotlib

DeprecationWarning: pyplot.hold is deprecated.

Future behavior will be consistent with the long-time default:

plot commands add elements without first clearing the

Axes and/or Figure.

plt.hold(b)

C:\Users\se\Anaconda3\lib\site-packages\matplotlib__init__.py:917: UserWarning: axe

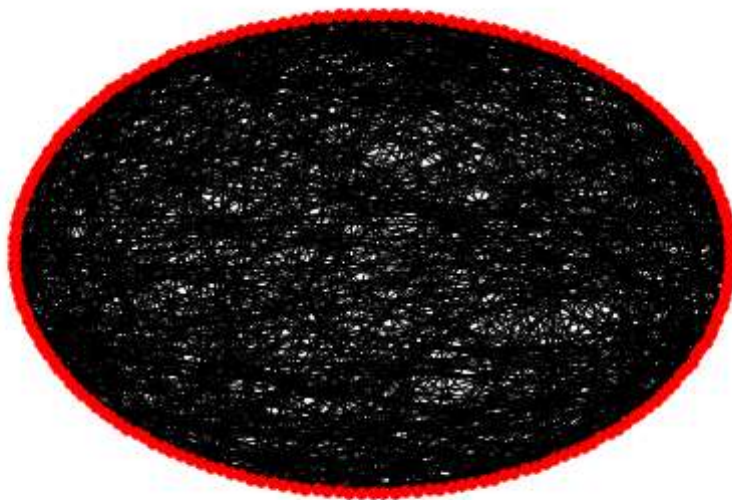
s.hold is deprecated. Please remove it from your matplotlibrc and/or style files.

warnings.warn(self.msg_depr_set % key)

C:\Users\se\Anaconda3\lib\site-packages\matplotlib\rcsetup.py:152: UserWarning: axe

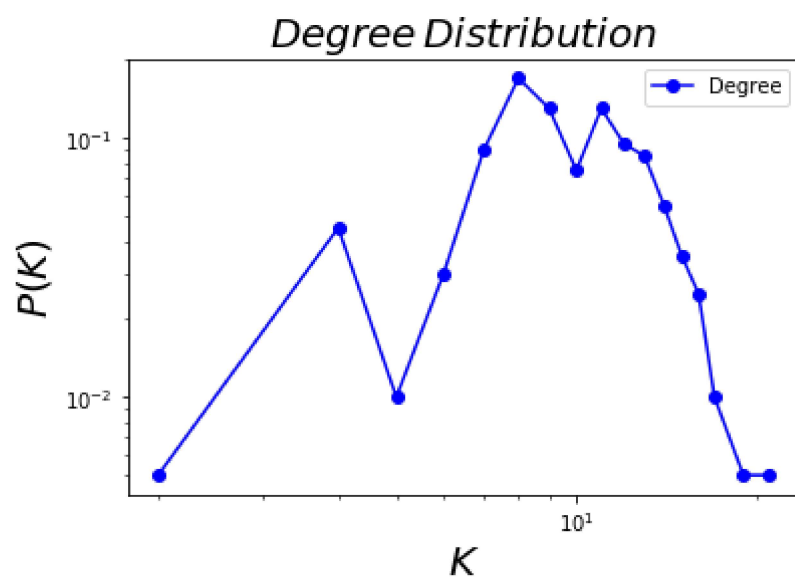
s.hold is deprecated, will be removed in 3.0

warnings.warn("axes.hold is deprecated, will be removed in 3.0")



In [29]:

```
plotDegreeDistribution(ER)
```



In [30]:

```
import networkx as nx
import matplotlib.pyplot as plt
WS = nx.random_graphs.watts_strogatz_graph(200, 4, 0.3) #生成包含200个节点、每个节点4个近邻、随机化重
pos = nx.circular_layout(WS) #定义一个布局, 此处采用了circular布局方式
nx.draw(WS, pos, with_labels=False, node_size = 30) #绘制图形
plt.show()
```

C:\Users\se\Anaconda3\lib\site-packages\networkx\drawing\nx_pylab.py:126: Matplotlib

DeprecationWarning: pyplot.hold is deprecated.

Future behavior will be consistent with the long-time default:

plot commands add elements without first clearing the

Axes and/or Figure.

b = plt.ishold()

C:\Users\se\Anaconda3\lib\site-packages\networkx\drawing\nx_pylab.py:138: Matplotlib

DeprecationWarning: pyplot.hold is deprecated.

Future behavior will be consistent with the long-time default:

plot commands add elements without first clearing the

Axes and/or Figure.

plt.hold(b)

C:\Users\se\Anaconda3\lib\site-packages\matplotlib__init__.py:917: UserWarning: axe

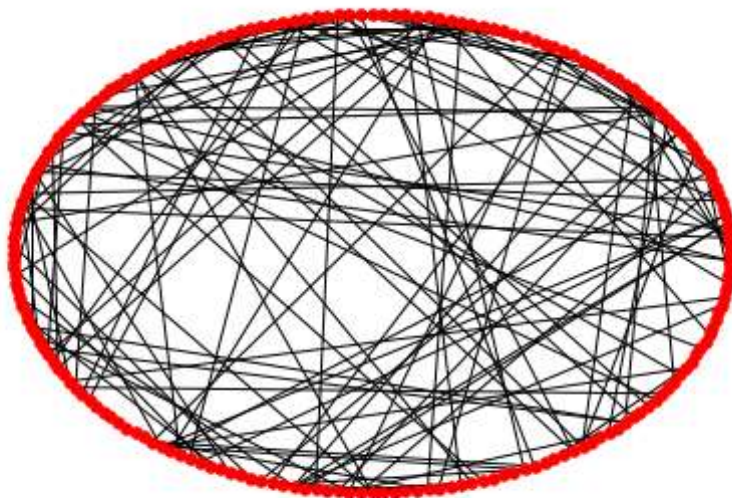
s.hold is deprecated. Please remove it from your matplotlibrc and/or style files.

warnings.warn(self.msg_depr_set % key)

C:\Users\se\Anaconda3\lib\site-packages\matplotlib\rcsetup.py:152: UserWarning: axe

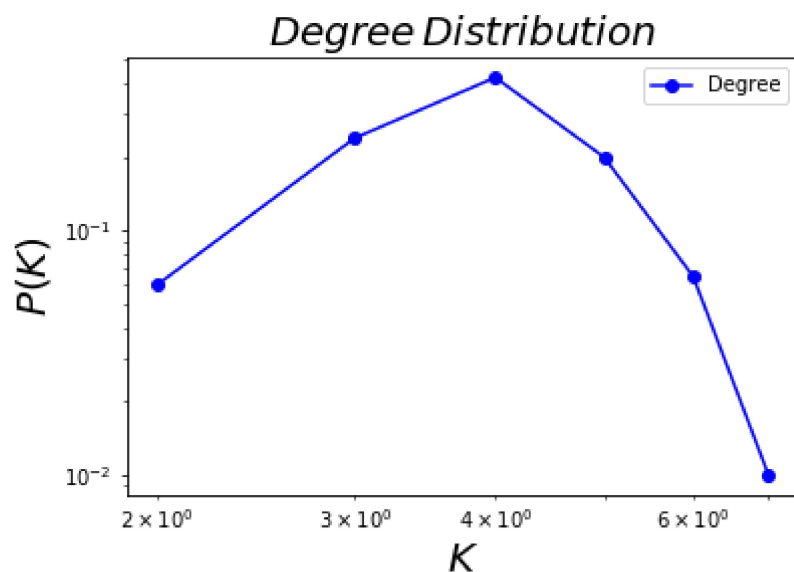
s.hold is deprecated, will be removed in 3.0

warnings.warn("axes.hold is deprecated, will be removed in 3.0")



In [31]:

```
plotDegreeDistribution(Ws)
```



In [32]:

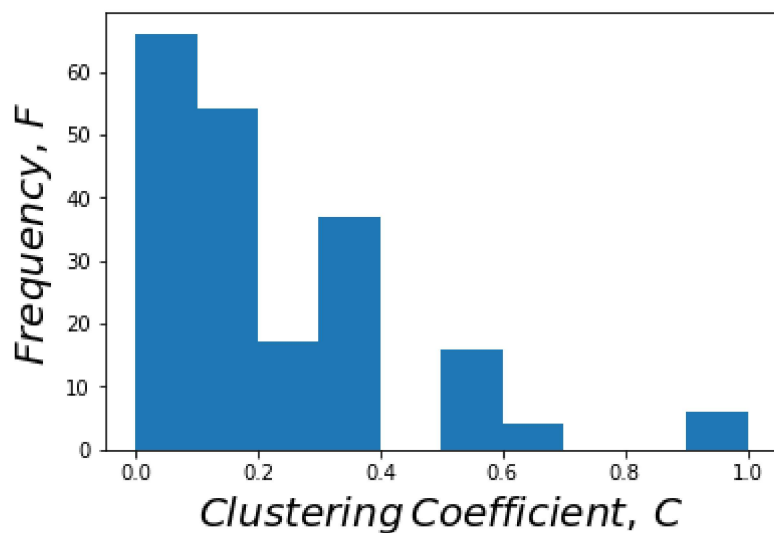
```
nx.diameter(Ws)
```

Out[32]:

9

In [33]:

```
cc = nx.clustering(Ws)
plt.hist(list(cc.values()), bins = 10)
plt.xlabel('Clustering Coefficient, C', fontsize = 20)
plt.ylabel('Frequency, F', fontsize = 20)
plt.show()
```



In [34]:

```
import numpy as np
np.mean(list(cc.values()))
```

Out[34]:

0.20478571428571427

In [35]:

```
import networkx as nx
import matplotlib.pyplot as plt
BA= nx.random_graphs.barabasi_albert_graph(200,2)  #生成n=20、m=1的BA无标度网络
pos = nx.spring_layout(BA)  #定义一个布局, 此处采用了spring布局方式
nx.draw(BA, pos, with_labels=False, node_size = 30)  #绘制图形
plt.show()
```

C:\Users\se\Anaconda3\lib\site-packages\networkx\drawing\nx_pylab.py:126: Matplotlib DeprecationWarning: pyplot.hold is deprecated.

Future behavior will be consistent with the long-time default:
plot commands add elements without first clearing the
Axes and/or Figure.

b = plt.ishold()

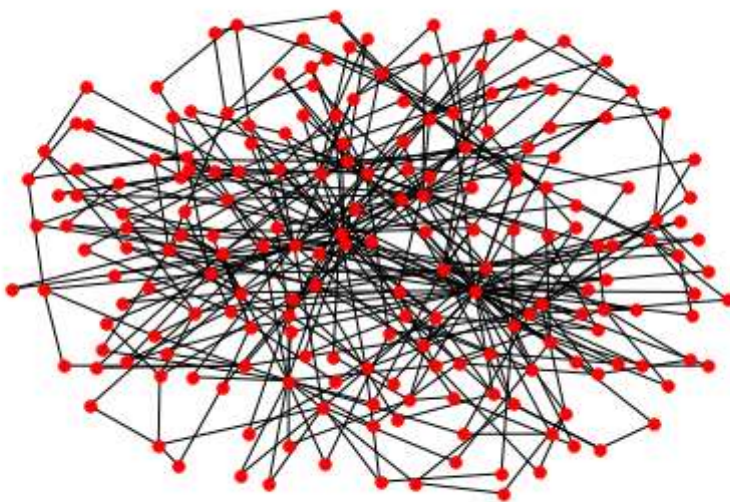
C:\Users\se\Anaconda3\lib\site-packages\networkx\drawing\nx_pylab.py:138: Matplotlib DeprecationWarning: pyplot.hold is deprecated.

Future behavior will be consistent with the long-time default:
plot commands add elements without first clearing the
Axes and/or Figure.

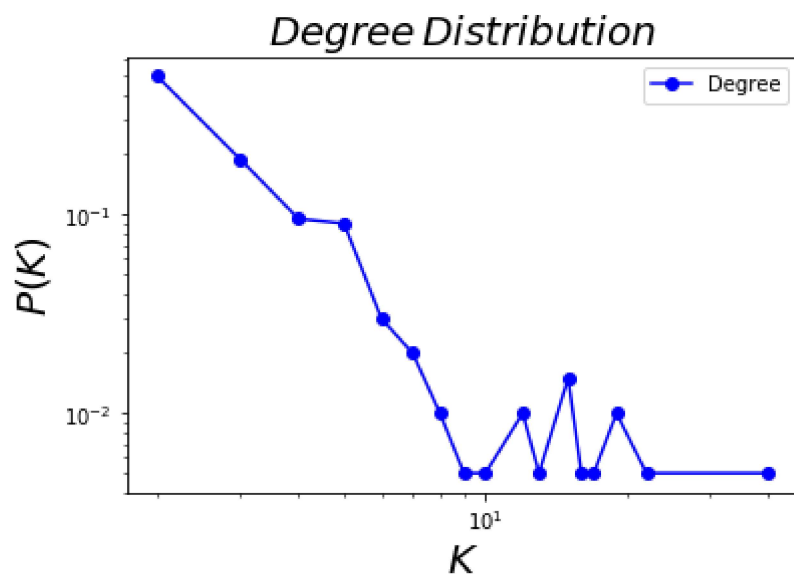
plt.hold(b)

C:\Users\se\Anaconda3\lib\site-packages\matplotlib__init__.py:917: UserWarning: axe s.hold is deprecated. Please remove it from your matplotlibrc and/or style files.
warnings.warn(self.msg_depr_set % key)

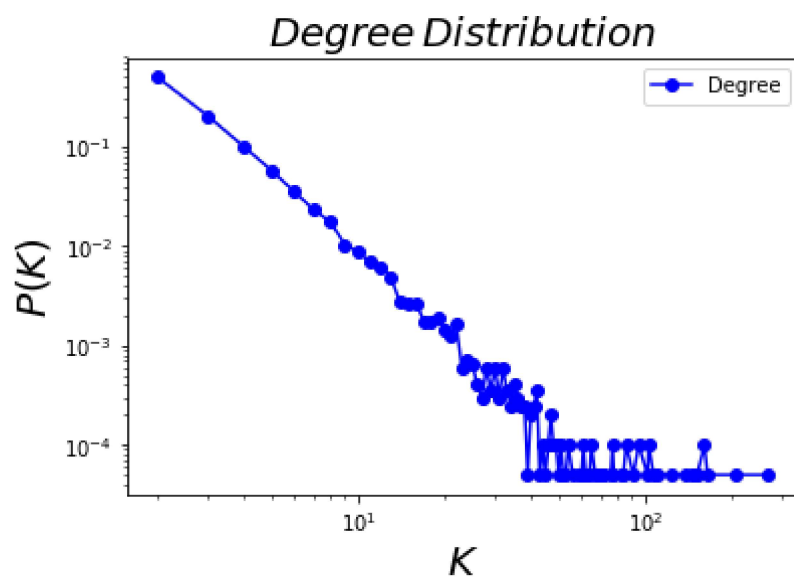
C:\Users\se\Anaconda3\lib\site-packages\matplotlib\rcsetup.py:152: UserWarning: axe s.hold is deprecated, will be removed in 3.0
warnings.warn("axes.hold is deprecated, will be removed in 3.0")



```
plotDegreeDistribution(BA)
```



```
BA= nx.random_graphs.barabasi_albert_graph(20000,2)  #生成n=20、m=1的BA无标度网络
plotDegreeDistribution(BA)
```



In [38]:

```
import networkx as nx
import matplotlib.pyplot as plt
BA= nx.random_graphs.barabasi_albert_graph(500,1)  #生成n=20、m=1的BA无标度网络
pos = nx.spring_layout(BA)  #定义一个布局，此处采用了spring布局方式
nx.draw(BA, pos, with_labels=False, node_size = 30)  #绘制图形
plt.show()
```

C:\Users\se\Anaconda3\lib\site-packages\networkx\drawing\nx_pylab.py:126: Matplotlib DeprecationWarning: pyplot.hold is deprecated.

Future behavior will be consistent with the long-time default:
plot commands add elements without first clearing the
Axes and/or Figure.

b = plt.ishold()

C:\Users\se\Anaconda3\lib\site-packages\networkx\drawing\nx_pylab.py:138: Matplotlib DeprecationWarning: pyplot.hold is deprecated.

Future behavior will be consistent with the long-time default:
plot commands add elements without first clearing the
Axes and/or Figure.

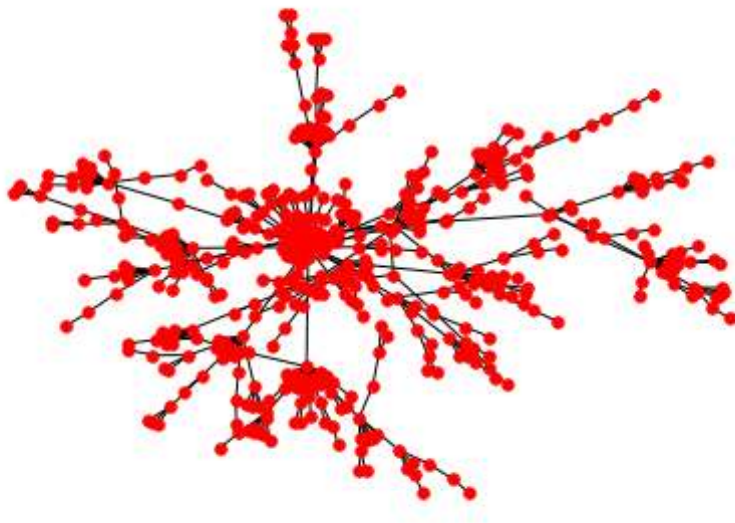
plt.hold(b)

C:\Users\se\Anaconda3\lib\site-packages\matplotlib__init__.py:917: UserWarning: axes.hold is deprecated. Please remove it from your matplotlibrc and/or style files.

warnings.warn(self.msg_depr_set % key)

C:\Users\se\Anaconda3\lib\site-packages\matplotlib\rcsetup.py:152: UserWarning: axes.hold is deprecated, will be removed in 3.0

warnings.warn("axes.hold is deprecated, will be removed in 3.0")



In [39]:

```
nx.degree_histogram(BA)[:3]
```

Out[39]:

```
[0, 330, 90]
```

In [40]:

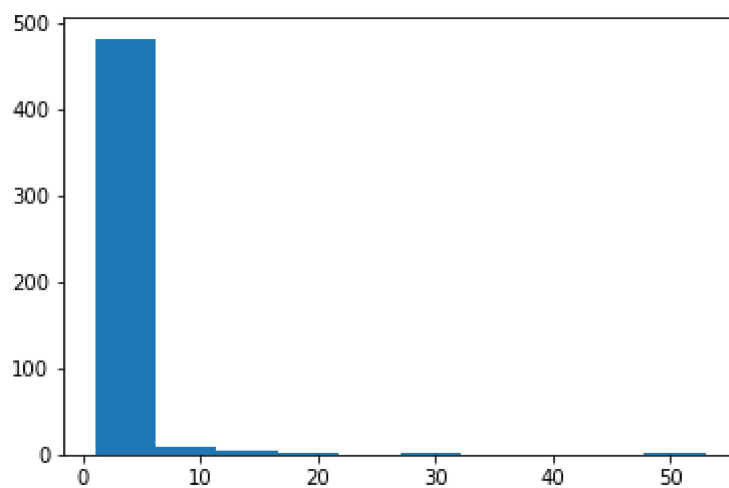
```
list(BA.degree().items())[:3]
```

Out[40]:

```
[(0, 28), (1, 53), (2, 11)]
```

In [41]:

```
plt.hist(list(BA.degree().values()))  
plt.show()
```



In [42]:

```
from collections import defaultdict
import numpy as np
def plotDegreeDistributionLongTail(G):
    degs = defaultdict(int)
    for i in G.degree().values(): degs[i]+=1
    items = sorted ( degs.items () )
    x, y = np.array(items).T
    y_sum = np.sum(y)
    y = [float(i)/y_sum for i in y]
    plt.plot(x, y, 'b-o')
    plt.legend(['Degree'])
    plt.xlabel('$K$', fontsize = 20)
    plt.ylabel('$P_K$', fontsize = 20)
    plt.title('$Degree\,Distribution$', fontsize = 20)
    plt.show()
```

```
BA= nx.random_graphs.barabasi_albert_graph(5000, 2) #生成n=20、m=1的BA无标度网络
plotDegreeDistributionLongTail(BA)
```

