In [ ]:
```
import networkx as nx
```

In [ ]:
```
import matplotlib.pyplot as plt
```

In [ ]:
```
G=nx.Graph()
```

In [ ]:
```
G.add_nodes_from(['a','b','c','d'])
```

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

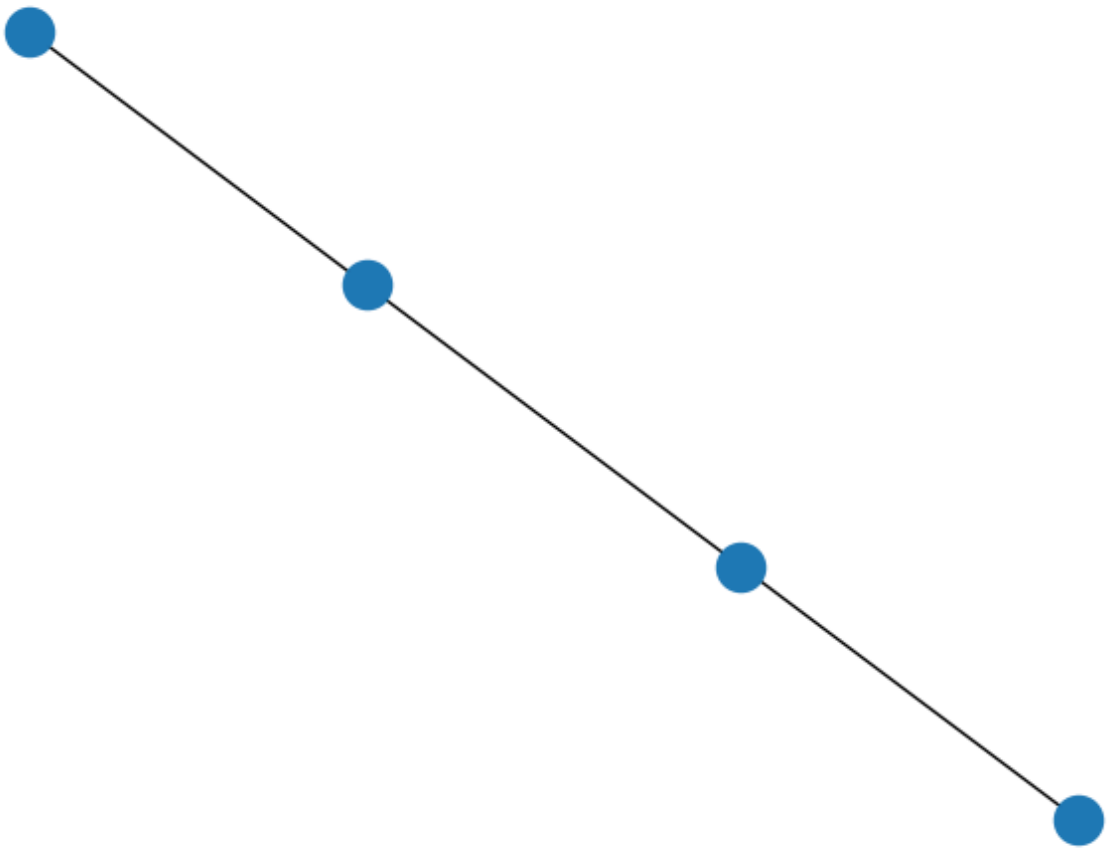
Out [5]:
```
NodeView(('a', 'b', 'c', 'd'))
```

In [ ]:
```
G.add_edge('a','b')
G.add_edge('a','c')
G.add_edge('b','d')
```

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

Out [7]:
```
EdgeView([('a', 'b'), ('a', 'c'), ('b', 'd')])
```

In [ ]:
```
nx.draw(G)
plt.show()
```



In [ ]:
```
nx.degree(G)
```

Out [9]:
```
DegreeView({'a': 2, 'b': 2, 'c': 1, 'd': 1})
```

In [ ]:
```
nx.degree_centrality(G)
```

Out [10]:
```
{'a': 0.6666666666666666,
 'b': 0.6666666666666666,
 'c': 0.3333333333333333,
 'd': 0.3333333333333333}
```

In [ ]:
```
nx.shortest_path(G,'a','d')
```

Out [11]:
```
['a', 'b', 'd']
```

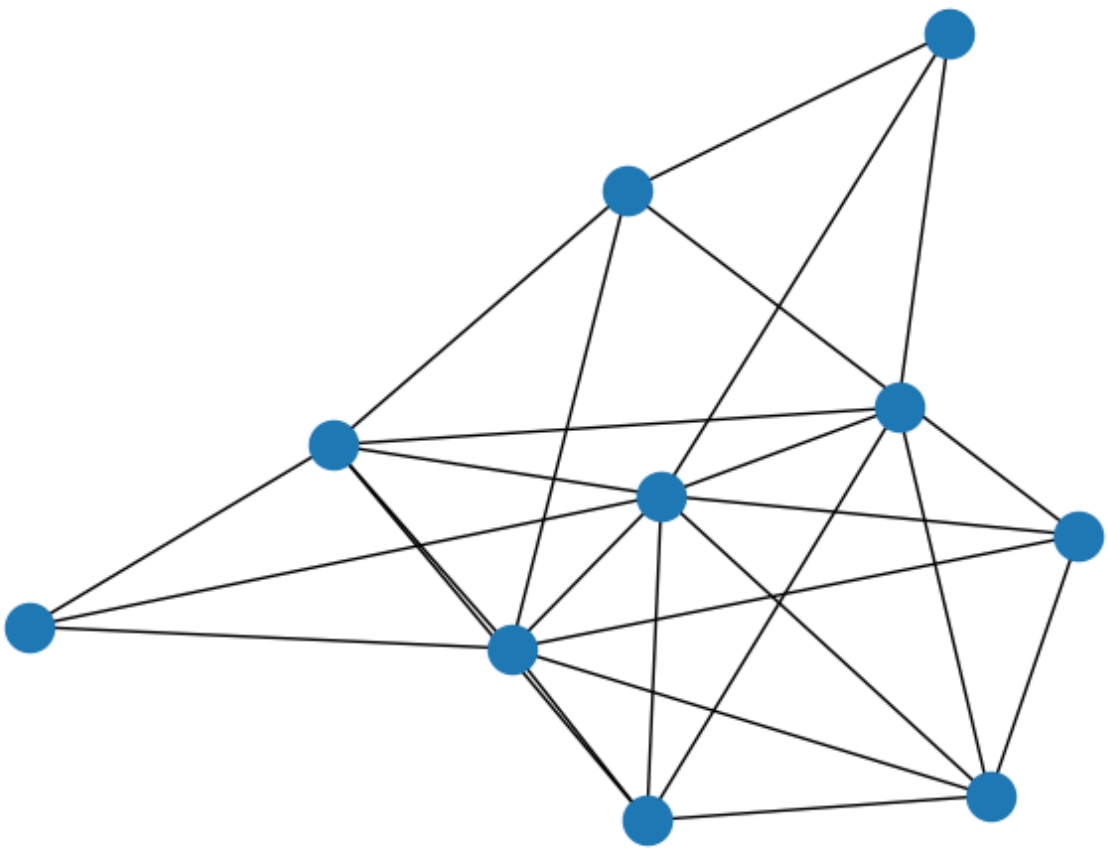
In [ ]:
```
nx.betweenness_centrality(G)
```

Out [12]:
```
{'a': 0.6666666666666666, 'b': 0.6666666666666666, 'c': 0.0, 'd': 0.0}
```

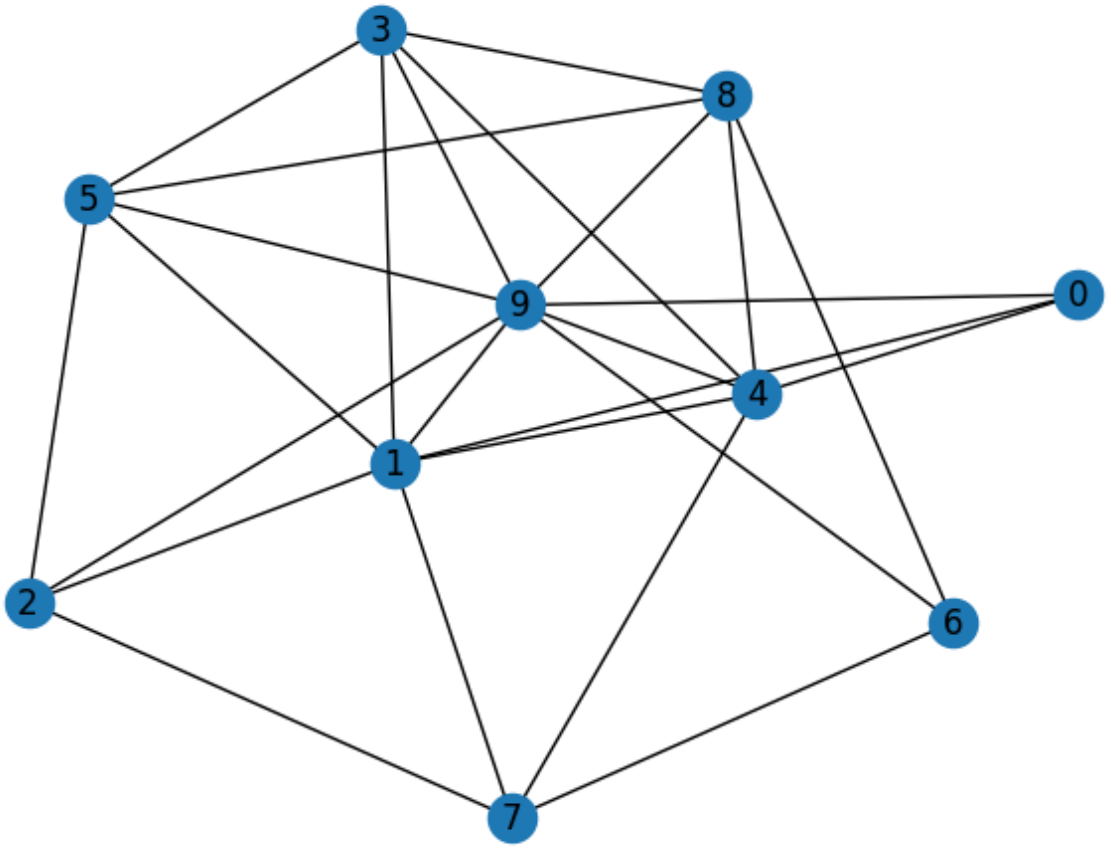
In [ ]:
```
nx.eigenvector_centrality(G)
```

Out [13]:
```
{'a': 0.6015008315175004,
 'b': 0.6015008315175004,
 'c': 0.37174823427120085,
 'd': 0.37174823427120085}
```

In [ ]:
```
G=nx.gnp_random_graph(10,0.5)
nx.draw(G)
```



In [ ]:
```
nx.draw(G,with_labels=1)
plt.show()
```



In [ ]:
```
nx.degree(G)
```

Out [16]:
```
DegreeView({0: 3, 1: 7, 2: 4, 3: 5, 4: 6, 5: 5, 6: 3, 7: 4, 8: 5, 9: 8})
```

In [ ]:
```
nx.degree_centrality(G)
```

Out [17]:
```
{0: 0.3333333333333333,
 1: 0.7777777777777777,
 2: 0.4444444444444444,
 3: 0.5555555555555556,
 4: 0.6666666666666666,
 5: 0.5555555555555556,
 6: 0.3333333333333333,
 7: 0.4444444444444444,
 8: 0.5555555555555556,
 9: 0.8888888888888888}
```

In [ ]:
```
nx.betweenness_centrality(G)
```

Out [18]:
```
{0: 0.0,
 1: 0.1111111111111109,
 2: 0.020833333333333332,
 3: 0.013888888888888888,
 4: 0.07870370370370369,
 5: 0.030092592592592587,
 6: 0.020833333333333332,
 7: 0.04629629629629629,
 8: 0.043981481481481476,
 9: 0.1898148148148148}
```

In [ ]:
```
nx.eigenvector_centrality(G)
```

Out [19]:
```
{0: 0.22336931310392164,
 1: 0.40240761612533166,
 2: 0.2581449509480931,
 3: 0.34007533231345144,
 4: 0.35906860930379075,
```

```
5: 0.3242929714596193,
6: 0.18003061170169088,
7: 0.222377614815088,
8: 0.30529999988859013,
9: 0.4435280159378195}
```

In [ ]:
```
m_influential=nx.degree_centrality(G)
```

In [ ]:
```
for w in sorted(m_influential,key=m_influential.get,reverse=True):
  print(w,m_influential[w])
```

```
9 0.8888888888888888
1 0.7777777777777777
4 0.6666666666666666
3 0.5555555555555556
5 0.5555555555555556
8 0.5555555555555556
2 0.4444444444444444
7 0.4444444444444444
0 0.3333333333333333
6 0.3333333333333333
```

In [ ]:
```
M=nx.Graph()
```

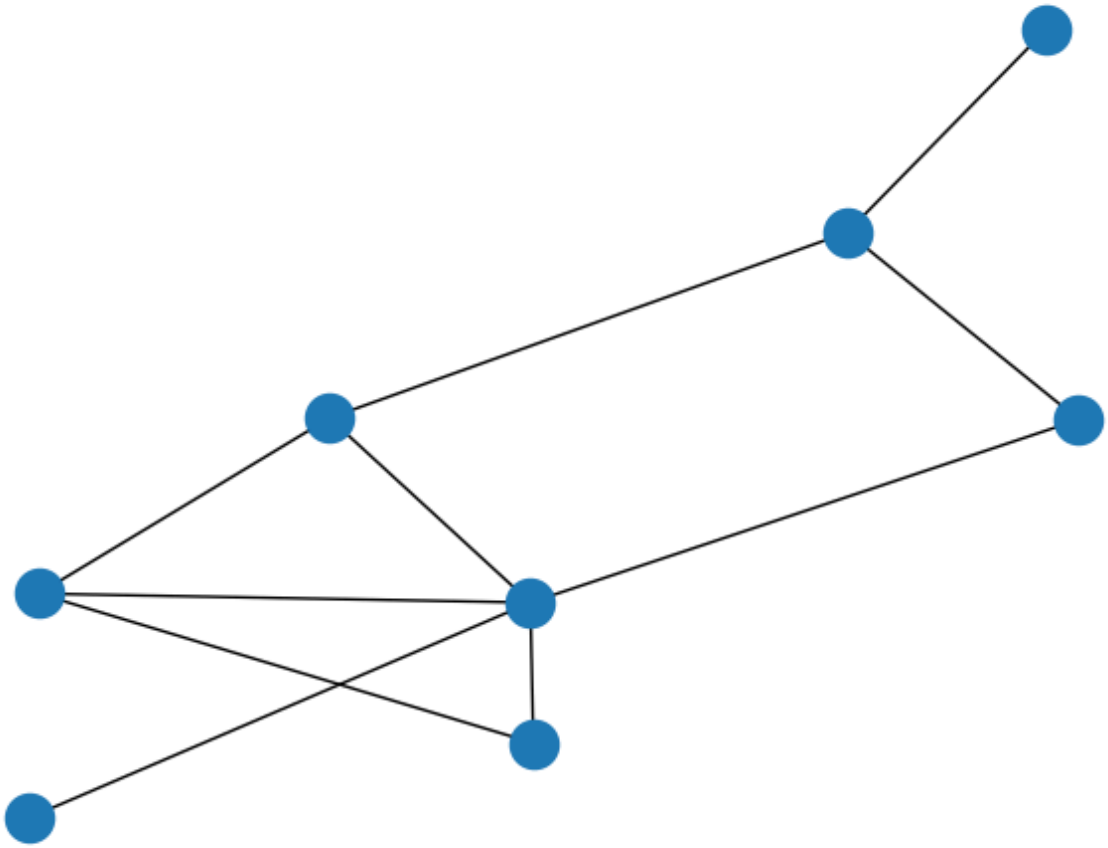
In [ ]:
```
M.add_nodes_from(['a','b','c','d','e','f','g','h'])
```

In [ ]:
```
M.nodes()
```

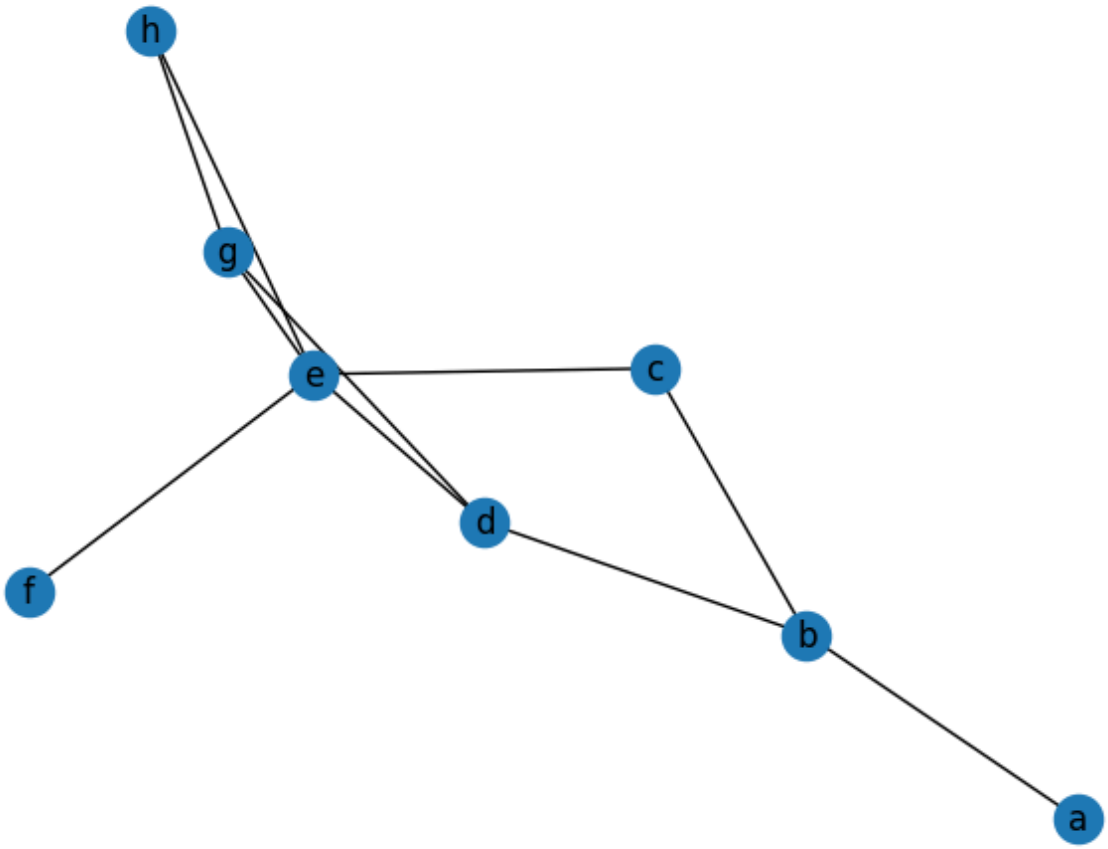
Out [24]:
```
NodeView(('a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'))
```

In [ ]:
```
M.add_edge('a','b')
M.add_edge('b','c')
M.add_edge('b','d')
M.add_edge('c','e')
M.add_edge('d','e')
M.add_edge('e','h')
M.add_edge('e','f')
M.add_edge('e','g')
M.add_edge('d','g')
M.add_edge('g','h')
```

In [ ]:
```
nx.draw(M)
plt.show()
```



In [ ]:
```
nx.draw(M,with_labels='a')
plt.show()
```



In [ ]:
```
nx.degree(M)
```

Out [28]:
```
DegreeView({'a': 1, 'b': 3, 'c': 2, 'd': 3, 'e': 5, 'f': 1, 'g': 3, 'h': 2})
```

In [ ]:
```
nx.degree_centrality(M)
```

Out [29]:
```
{'a': 0.14285714285714285,
 'b': 0.42857142857142855,
 'c': 0.2857142857142857,
 'd': 0.42857142857142855,
 'e': 0.7142857142857142,
 'f': 0.14285714285714285,
 'g': 0.42857142857142855,
 'h': 0.2857142857142857}
```

In [ ]:
```
nx.betweenness_centrality(M)
```

Out [30]:
```
{'a': 0.0,
 'b': 0.30952380952380953,
 'c': 0.12698412698412698,
 'd': 0.25396825396825395,
 'e': 0.492063492063492,
 'f': 0.0,
 'g': 0.055555555555555546,
 'h': 0.0}
```

In [ ]:
```
nx.eigenvector_centrality(M)
```

Out [31]:
```
{'a': 0.0892314798120002,
 'b': 0.2660740286403953,
 'c': 0.27754341242731234,
 'd': 0.4266201277775895,
 'e': 0.5615233897735086,
 'f': 0.18831193261533638,
 'g': 0.4445292724700449,
 'h': 0.33738864796561363}
```

In [ ]:
```
m_influential=nx.betweenness_centrality(M)
```

In [ ]:
```
for x in sorted(m_influential,key=m_influential.get,reverse=True):
  print(x,m_influential[x])
```

```
e 0.492063492063492
b 0.30952380952380953
d 0.25396825396825395
c 0.12698412698412698
g 0.055555555555555546
a 0.0
f 0.0
h 0.0
```

In [ ]:
```
cliques=list(nx.find_cliques(M))
cliques
```

Out [34]:
```
[['b', 'd'],
 ['b', 'c'],
 ['b', 'a'],
 ['e', 'f'],
 ['e', 'g', 'h'],
 ['e', 'g', 'd'],
 ['e', 'c']]
```

In [ ]:

In [ ]:

In [ ]: