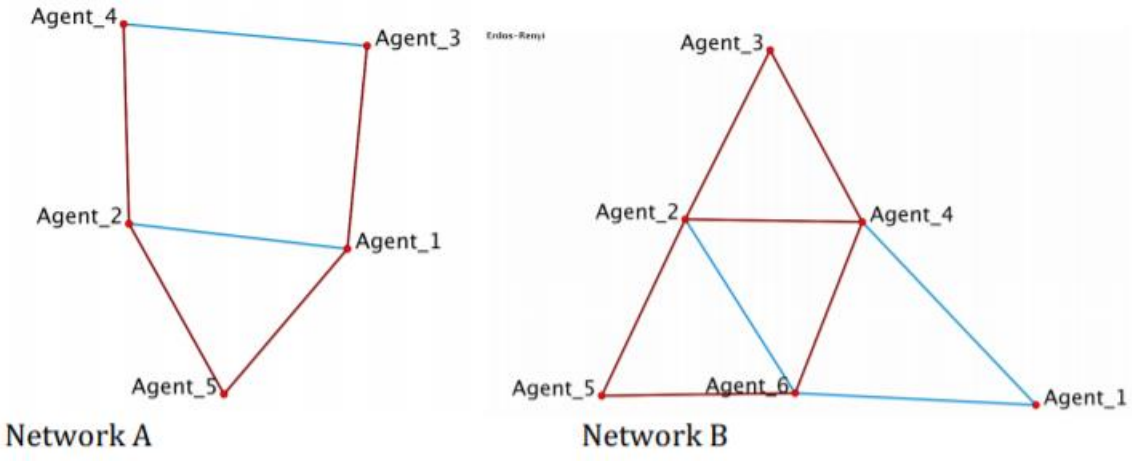
Assignment 3

Guan Yue Wang



1 Adjacency Matrix

Network A

	1	2	3	4	5
1	0	1	1	0	1
2	1	0	0	1	1
3	1	0	0	1	0
4	0	1	1	0	0
5	1	1	0	0	0

Network B

	1	2	3	4	5	6
1	0	0	0	1	0	1
2	0	0	1	1	1	1
3	0	1	0	1	0	0
4	1	1	1	0	0	1
5	0	1	0	0	0	1
6	1	1	0	1	1	0

2 Degree Centralities

Network A

	1	2	3	4	5	Sum	n-1	**Degree**
1	0	1	1	0	1	3	4	**3/4**
2	1	0	0	1	1	3	4	**3/4**
3	1	0	0	1	0	2	4	**2/4**
4	0	1	1	0	0	2	4	**2/4**
5	1	1	0	0	0	2	4	**2/4**

Network B

	1	2	3	4	5	6	Sum	n-1	**Degree**
1	0	0	0	1	0	1	2	5	**2/5**
2	0	0	1	1	1	1	4	5	**4/5**
3	0	1	0	1	0	0	2	5	**2/5**
4	1	1	1	0	0	1	4	5	**4/5**
5	0	1	0	0	0	1	2	5	**2/5**
6	1	1	0	1	1	0	4	5	**4/5**

3 Geodesic

Network A

From	To	Geodesic
1	2	(1,2)
1	3	(1,3)
1	4	(1,2,4) (1,3,4)
1	5	(1,5)
2	3	(2,1,3) (2,4,3)
2	4	(2,4)
2	5	(2,5)
3	4	(3,4)
3	5	(3,1,5)
4	5	(4,2,5)

Network B

From	To	Geodesic
1	2	(1,4,2) (1,6,2)
1	3	(1,4,3)
1	4	(1,4)
1	5	(1,6,5)
1	6	(1,6)
2	3	(2,3)
2	4	(2,4)
2	5	(2,5)
2	6	(2,6)
3	4	(3,4)
3	5	(3,2,5)
3	6	(3,2,6) (3,4,6)
4	5	(4,2,5) (4,6,5)
4	6	(4,6)
5	6	(5,6)

4 Betweenness

Network A

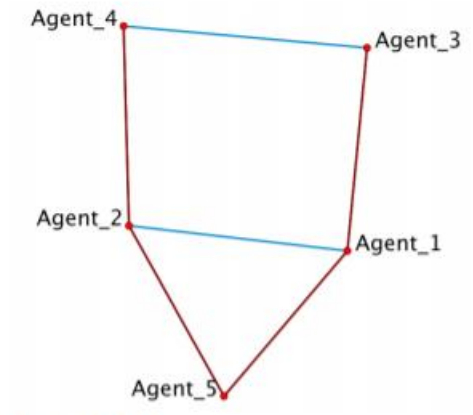
From	To	Geodesic	1	2	3	4	5	
1	2	(1,2)	0	0	0	0	0	
1	3	(1,3)	0	0	0	0	0	
1	4	(1,2,4) (1,3,4)	0	0.5	0.5	0	0	
1	5	(1,5)	0	0	0	0	0	
2	3	(2,1,3) (2,4,3)	0.5	0	0	0.5	0	
2	4	(2,4)	0	0	0	0	0	
2	5	(2,5)	0	0	0	0	0	
3	4	(3,4)	0	0	0	0	0	
3	5	(3,1,5)	1	0	0	0	0	
4	5	(4,2,5)	0	1	0	0	0	
		Sum	1.5	1.5	0.5	0.5	0	Numerator
		(n-1)(n-2)/2 = 6	6	6	6	6	6	Denominator
		Betweenness	0.25	0.25	0.08	0.08	0	Betweenness

Network B

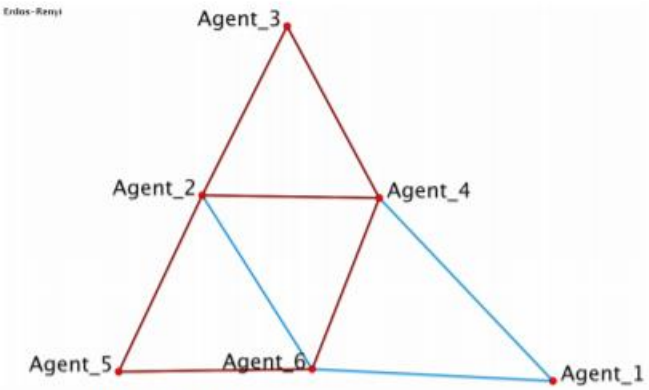
From	To	Geodesic
1	2	(1,4,2) (1,6,2)
1	3	(1,4,3)
1	4	(1,4)
1	5	(1,6,5)
1	6	(1,6)
2	3	(2,3)
2	4	(2,4)
2	5	(2,5)
2	6	(2,6)
3	4	(3,4)
3	5	(3,2,5)
3	6	(3,2,6) (3,4,6)
4	5	(4,2,5) (4,6,5)
4	6	(4,6)
5	6	(5,6)

From	To	Geodesic	1	2	3	4	5	6	
1	2	(1,4,2) (1,6,2)	0	0	0	0.5	0	0.5	
1	3	(1,4,3)	0	0	0	1	0	0	
1	4	(1,4)	0	0	0	0	0	0	
1	5	(1,6,5)	0	0	0	0	0	1	
1	6	(1,6)	0	0	0	0	0	0	
2	3	(2,3)	0	0	0	0	0	0	
2	4	(2,4)	0	0	0	0	0	0	
2	5	(2,5)	0	0	0	0	0	0	
2	6	(2,6)	0	0	0	0	0	0	
3	4	(3,4)	0	0	0	0	0	0	
3	5	(3,2,5)	0	1	0	0	0	0	
3	6	(3,2,6) (3,4,6)	0	0.5	0	0.5	0	0	
4	5	(4,2,5) (4,6,5)	0	0.5	0	0	0	0.5	
4	6	(4,6)	0	0	0	0	0	0	
5	6	(5,6)	0	0	0	0	0	0	
		Sum	0	2	0	2	0	2	Numerator
		(n-1)(n-2)/2 = 10	10	10	10	10	10	10	Denominator
		Betweenness	0	0.2	0	0.2	0	0.2	Betweenness

5. Closeness



Network A



Network B

Network A

	1	2	3	4	5	Sum	n-1	**closeness (n-1)/sum**
1		1	1	2	1	5	4	**4/5**
2	1		2	1	1	5	4	**4/5**
3	1	2		1	2	6	4	**4/6**
4	2	1	1		2	6	4	**4/6**
5	1	1	2	2		6	4	**4/6**

Network B

	1	2	3	4	5	6	Sum	n-1	**closeness (n-1)/sum**
1		2	2	1	2	1	8	5	**5/8**
2	2		1	1	1	1	6	5	**5/6**
3	2	1		1	2	2	8	5	**5/8**
4	1	1	1		2	1	6	5	**5/6**
5	2	1	2	2		1	8	5	**5/8**
6	1	1	2	1	1		6	5	**5/6**

6. Influence

Network A

Agent 1 and 2 have the greater influence as they have the greatest degree, betweenness, closeness centrality scores.

Network B

Agent 2, 4, 6 have the greater influence as they have the greatest degree, betweenness, closeness centrality scores.

7. Network Density

Network A

Possible links = 5 choose 2 = 5!/(2!3!) = 10

Actual edges = 6

Network Density = 6/10 = 60%

Network B

Possible links = 6 choose 2 = 6!/(2!4!) = 15

Actual edges = 9

Network Density = 9/15 = 60%

8. Diameter

Diameter is based on the longest geodesic in the network.

Referring to question 3, we know both network A and B have diameter 3

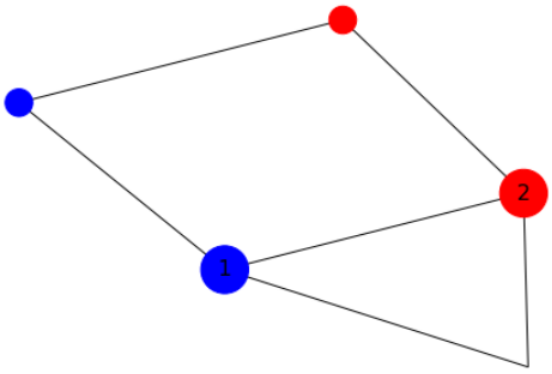
Python Code and Output (actual files are attached as additional documents):

In [1]:
```
import networkx as nx
```

In [3]:
```
# Network A
# Build Network
networkA = nx.Graph()
networkA.add_edges_from([(1,2), (1,3),  (2,4), (1,5),(2,5), (3,4)])

# Validations
print ("Network A")
print ("Degree Centrality: ", nx.degree_centrality(networkA))
print ("Betweenness Centrality: ", nx.betweenness_centrality(networkA))
print ("Closeness Centrality: ", nx.closeness_centrality(networkA))
print ("Modified Graph of Network A:")

# Graph Modification
bcA = nx.betweenness_centrality(networkA)
degreeA = nx.degree_centrality(networkA)


nodecolorA = []
nodesizeA = []
labelsA={}
posA=nx.spring_layout(networkA)

for v in nx.nodes(networkA):
    if v % 2 == 0:
        nodecolorA.append("red")
    else:
        nodecolorA.append("blue")

for v in bcA.values():
    nodesizeA.append(v*5000)

for i in nx.nodes(networkA):
    if degreeA[i] == max(degreeA.values()):
        labelsA[i] = i

nx.draw(networkA, posA, nodelist=nx.nodes(networkA), node_size=nodesizeA, node_color = nodecolorA)
nx.draw_networkx_labels(networkA,posA,labelsA,font_size=16)
```

```
Network A
Degree Centrality:  {1: 0.75, 2: 0.75, 3: 0.5, 4: 0.5, 5: 0.5}
Betweenness Centrality:  {1: 0.25, 2: 0.25, 3: 0.08333333333333333, 4: 0.08333333333333333, 5: 0.0}
Closeness Centrality:  {1: 0.8, 2: 0.8, 3: 0.6666666666666666, 4: 0.6666666666666666, 5: 0.6666666666666666}
Modified Graph of Network A:
```

Out[3]:
```
{1: Text(-0.27654735707049444, -0.44806753664863336, '1'),
 2: Text(0.5264535965290095, -0.015633377934734134, '2')}
```



In [4]:
```
# Network B
# Build Network
networkB = nx.Graph()
networkB.add_edges_from([(1,6), (1,4), (2,3), (2,4), (2,5), (2,6),(2,3), (2,4), (2,5), (3,4),(4,6),(5,6)])
# Validations
print ("Network B")
print ("Degree Centrality: ", nx.degree_centrality(networkB))
print ("Betweenness Centrality: ", nx.betweenness_centrality(networkB))
print ("Closeness Centrality: ", nx.closeness_centrality(networkB))
print ("Modified Graph of Network B:")

# Graph Modification
bcB = nx.betweenness_centrality(networkB)
degreeB = nx.degree_centrality(networkB)

nodecolorB = []
nodesizeB = []
labelsB={}
posB=nx.spring_layout(networkB)


for v in nx.nodes(networkB):
    if v % 2 == 0:
        nodecolorB.append("red")
    else:
        nodecolorB.append("blue")

for v in bcB.values():
    nodesizeB.append(v*5000)



for i in nx.nodes(networkB):
    if degreeB[i] == max(degreeB.values()):
        labelsB[i] = i

nx.draw(networkB, posB, nodelist=nx.nodes(networkB), node_size=nodesizeB, node_color = nodecolorB)
nx.draw_networkx_labels(networkB,posB,labelsB,font_size=16)
```

```
Network B
Degree Centrality:  {1: 0.4, 6: 0.8, 4: 0.8, 2: 0.8, 3: 0.4, 5: 0.4}
Betweenness Centrality:  {1: 0.0, 6: 0.2, 4: 0.2, 2: 0.2, 3: 0.0, 5: 0.0}
Closeness Centrality:  {1: 0.625, 6: 0.8333333333333334, 4: 0.8333333333333334, 2: 0.8333333333333334, 3: 0.625, 5: 0.625}
Modified Graph of Network B:
```

:
```
{6: Text(0.45514844694035683, -0.06426671333111683, '6'),
 4: Text(-0.28119873489661934, -0.3591229166437512, '4'),
 2: Text(-0.17280029591660806, 0.42344178298078267, '2')}
```