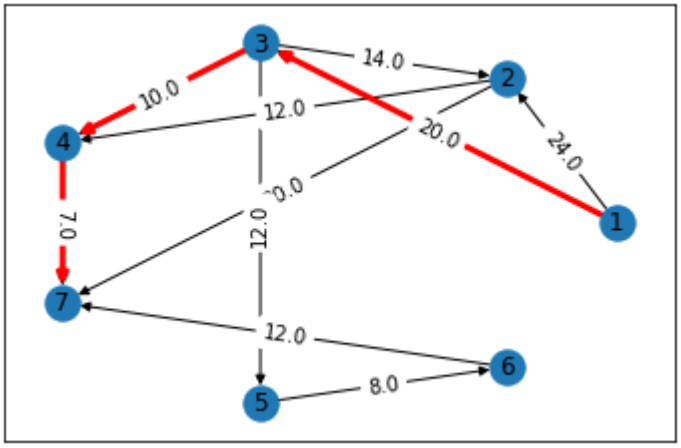
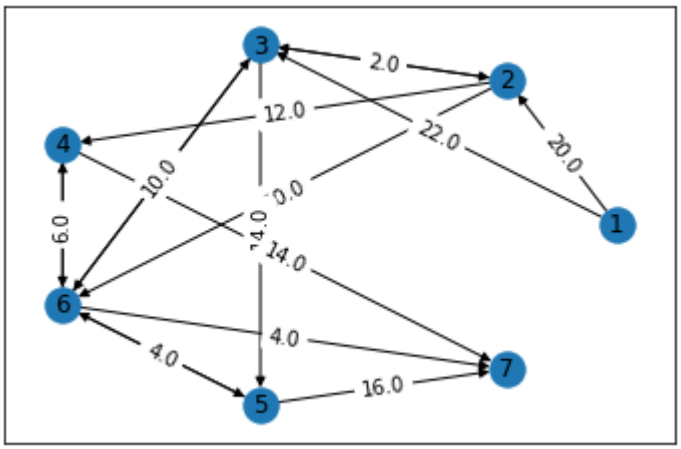
B [17]:
```
!pip install pulp
import networkx as nx
import matplotlib.pyplot as plt
from pulp import *

# 3
elist = [(1, 2, 24.0), (1, 3, 20.0), (3, 2, 14.0), (2, 4, 12.0), (2, 7, 30.0
         (5, 6, 8.0), (6, 7, 12.0)]
G = nx.DiGraph()
G.add_weighted_edges_from(elist)
weights = nx.get_edge_attributes(G, 'weight')
pos = nx.circular_layout(G)
X = nx.shortest_path(G, source=1, target=7, weight='weight')
edges = [(a, b) for a, b in zip(X, X[1:])]
nx.draw_networkx(G, pos=pos)
nx.draw_networkx_edge_labels(G, pos, edge_labels=weights)
nx.draw_networkx_edges(G, pos=pos, edgelist=edges, edge_color="r", width=3)
plt.show()
print(nx.shortest_path(G, source=1, target=7, weight='weight'))
print(nx.shortest_path_length(G, source=1, target=7, weight='weight'))
# 1
G = nx.DiGraph()
G.add_edge('1', '2', capacity=20.0)
G.add_edge('1', '3', capacity=22.0)
G.add_edge('2', '3', capacity=4.0)
G.add_edge('3', '2', capacity=2.0)
G.add_edge('2', '4', capacity=12.0)
G.add_edge('2', '6', capacity=10.0)
G.add_edge('3', '5', capacity=14.0)
G.add_edge('3', '6', capacity=10.0)
G.add_edge('4', '7', capacity=14.0)
G.add_edge('4', '6', capacity=6.0)
G.add_edge('5', '6', capacity=4.0)
G.add_edge('5', '7', capacity=16.0)
G.add_edge('6', '3', capacity=10.0)
G.add_edge('6', '4', capacity=6.0)
G.add_edge('6', '5', capacity=4.0)
G.add_edge('6', '7', capacity=4.0)
pos = nx.circular_layout(G)
weights = nx.get_edge_attributes(G, 'capacity')
nx.draw_networkx(G, pos)
nx.draw_networkx_edge_labels(G, pos, edge_labels=weights)
plt.show()
print(nx.maximum_flow(G, '1', '7'))
# 2
G = nx.DiGraph()
G.add_node('1', demand=-20)
G.add_node('4', demand=5)
G.add_node('5', demand=15)
G.add_edge('1', '2', weight=12, capacity=15)
G.add_edge('1', '3', weight=14, capacity=8)
G.add_edge('2', '3', weight=5, capacity=100)
G.add_edge('2', '4', weight=4, capacity=4)
G.add_edge('2', '5', weight=9, capacity=10)
G.add_edge('3', '4', weight=2, capacity=15)
G.add_edge('3', '5', weight=5, capacity=5)
```

```
G.add_edge('4', '5', weight=12, capacity=100)
G.add_edge('5', '3', weight=5, capacity=4)
print(nx.min_cost_flow(G))
print(nx.min_cost_flow_cost(G))
```

```
Requirement already satisfied: pulp in /srv/conda/envs/notebook/lib/pytho
n3.6/site-packages (2.4)
Requirement already satisfied: amply>=0.1.2 in /srv/conda/envs/notebook/l
ib/python3.6/site-packages (from pulp) (0.1.4)
Requirement already satisfied: pyparsing in /srv/conda/envs/notebook/lib/
python3.6/site-packages (from amply>=0.1.2->pulp) (2.4.7)
Requirement already satisfied: docutils>=0.3 in /srv/conda/envs/notebook/
lib/python3.6/site-packages (from amply>=0.1.2->pulp) (0.17.1)
```




```
[1, 3, 4, 7]
37.0
```




```
(34.0, {'1': {'2': 14.0, '3': 20.0}, '2': {'3': 0, '4': 8.0, '6': 6.0},
'3': {'2': 0, '5': 12.0, '6': 8.0}, '4': {'7': 14.0, '6': 0}, '6': {'3':
0, '4': 6.0, '5': 4.0, '7': 4.0}, '5': {'6': 0, '7': 16.0}, '7': {}})
{'1': {'2': 14, '3': 6}, '4': {'5': 0}, '5': {'3': 0}, '2': {'3': 0, '4':
4, '5': 10}, '3': {'4': 1, '5': 5}}
385
```

B [ ]:

B [ ]: