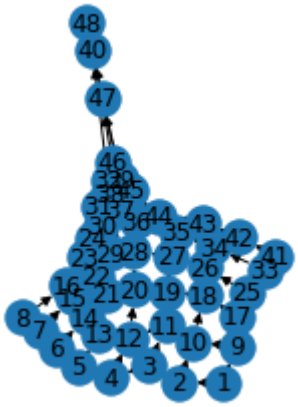
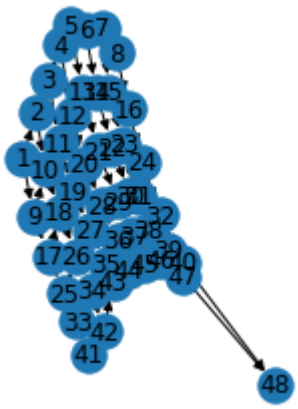
In [1]:
```
import matplotlib.pyplot as plt
import networkx as nx
import numpy as np
import random

##инициализация графа
e = [(1, 2, 16), (2, 3, 13), (3, 4, 17), (4, 5, 14), (5, 6, 17), (6, 7, 12),
     (1, 9, 14), (2, 10, 11), (3, 11, 18), (4, 12, 14), (5, 13, 20), (6, 14,
     (9, 10, 14), (10, 11, 20), (11, 12, 8), (12, 13, 12), (13, 14, 10), (14
     (9, 17, 18), (10, 18, 8), (11, 19, 20), (12, 20, 12), (13, 21, 17), (14
     (17, 18, 18), (18, 19, 15), (19, 20, 16), (20, 21, 8), (21, 22, 17), (2
     (17, 25, 11), (18, 26, 11), (19, 27, 8), (20, 28, 20), (21, 29, 15), (2
     (25, 26, 17), (26, 27, 18), (27, 28, 10), (28, 29, 13), (29, 30, 9), (3
     (25, 33, 14), (26, 34, 8), (27, 35, 10), (28, 36, 9), (29, 37, 9), (30,
     (33, 34, 12), (34, 35, 20), (35, 36, 11), (36, 37, 12), (37, 38, 19),
     (33, 41, 18), (34, 42, 11), (35, 43, 8), (36, 44, 8), (37, 45, 16), (38
     (41, 42, 20), (42, 43, 20), (43, 44, 14), (44, 45, 11), (45, 46, 17),
G = nx.DiGraph()
for fr, to, cp in e:
    G.add_edge(fr, to, weight=cp)

plt.subplot(121)
nx.draw(G, with_labels=True)
plt.show();

##Минимальный путь с помошью ф-ции библиотеки networkx.algorithms.shortest_p
p = nx.shortest_path(G, source=1, target=48, weight="weight", method="dijkst
weight = nx.path_weight(G, p, "weight")

print("Путь ", end="")
for count, value in enumerate(p):
    if count < len(p) - 1:
        print(value, end=" -> ")
    else:
        print(value)
print("Вес пути", weight)

##инициализация графа
e = [(1, 2, 19), (2, 3, 15), (3, 4, 23), (4, 5, 22), (5, 6, 16), (6, 7, 16),
     (1, 9, 23), (2, 10, 17), (3, 11, 23), (4, 12, 15), (5, 13, 15), (6, 14,
     (9, 10, 15), (10, 11, 16), (11, 12, 15), (12, 13, 21), (13, 14, 20), (1
     (9, 17, 23), (10, 18, 19), (11, 19, 19), (12, 20, 24), (13, 21, 16), (1
     (17, 18, 21), (18, 19, 23), (19, 20, 15), (20, 21, 15), (21, 22, 18),
     (17, 25, 23), (18, 26, 15), (19, 27, 18), (20, 28, 19), (21, 29, 16),
     (25, 26, 18), (26, 27, 20), (27, 28, 17), (28, 29, 20), (29, 30, 16),
     (25, 33, 23), (26, 34, 20), (27, 35, 18), (28, 36, 20), (29, 37, 23),
     (33, 34, 22), (34, 35, 22), (35, 36, 21), (36, 37, 24), (37, 38, 21),
     (33, 41, 19), (34, 42, 15), (35, 43, 20), (36, 44, 22), (37, 45, 21),
     (41, 42, 22), (42, 43, 15), (43, 44, 24), (44, 45, 24), (45, 46, 16),
G = nx.DiGraph()
for fr, to, cp in e:
    G.add_edge(fr, to, weight=cp)

plt.subplot(121)
nx.draw(G, with_labels=True)
plt.show();
```

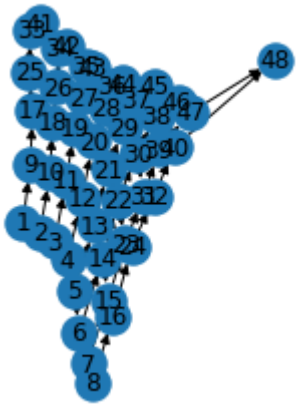
```
X = nx.dag_longest_path(G, weight='weight')
weight = nx.path_weight(G, X, "weight")

print("Путь ", end="")
for count, value in enumerate(X):
    if count < len(p) - 1:
        print(value, end=" -> ")
    else:
        print(value)
print("Вес пути", weight)

# 3

##инициализация графа
e = [(1, 2, 16), (2, 3, 13), (3, 4, 17), (4, 5, 14), (5, 6, 17), (6, 7, 12),
     (1, 9, 14), (2, 10, 11), (3, 11, 18), (4, 12, 14), (5, 13, 20), (6, 14,
     (9, 10, 14), (10, 11, 20), (11, 12, 8), (12, 13, 12), (13, 14, 10), (14
     (9, 17, 18), (10, 18, 8), (11, 19, 20), (12, 20, 12), (13, 21, 17), (14
     (17, 18, 18), (18, 19, 15), (19, 20, 16), (20, 21, 8), (21, 22, 17), (2
     (17, 25, 11), (18, 26, 11), (19, 27, 8), (20, 28, 20), (21, 29, 15), (2
     (25, 26, 17), (26, 27, 18), (27, 28, 10), (28, 29, 13), (29, 30, 9), (3
     (25, 33, 14), (26, 34, 8), (27, 35, 10), (28, 36, 9), (29, 37, 9), (30,
     (33, 34, 12), (34, 35, 20), (35, 36, 11), (36, 37, 12), (37, 38, 19),
     (33, 41, 18), (34, 42, 11), (35, 43, 8), (36, 44, 8), (37, 45, 16), (38
     (41, 42, 20), (42, 43, 20), (43, 44, 14), (44, 45, 11), (45, 46, 17),
G = nx.DiGraph()
for fr, to, cp in e:
    G.add_edge(fr, to, weight=np.random.uniform(18, 24))

plt.subplot(121)
nx.draw(G, with_labels=True)
plt.show();

##Минимальный путь с помошью ф-ции библиотеки networkx.algorithms.shortest_p
p = nx.shortest_path(G, source=1, target=48, weight="weight", method="dijkst
weight = nx.path_weight(G, p, "weight")
print('Задание 3')
print("Путь ", end="")
for count, value in enumerate(p):
    if count < len(p) - 1:
        print(value, end=" -> ")
    else:
        print(value)
print("Вес пути", weight)
```



```
Путь 1 -> 2 -> 10 -> 18 -> 19 -> 27 -> 28 -> 29 -> 30 -> 31 -> 39 -> 40 -
> 48
Вес пути 131
```



```
Путь 1 -> 9 -> 17 -> 25 -> 33 -> 34 -> 35 -> 36 -> 37 -> 38 -> 39 -> 47 -
> 48
Вес пути 267
```



```
Задание 3
Путь 1 -> 2 -> 10 -> 11 -> 12 -> 20 -> 21 -> 29 -> 30 -> 38 -> 39 -> 47 -
> 48
Вес пути 241.9518324549076
```

In [ ]: