

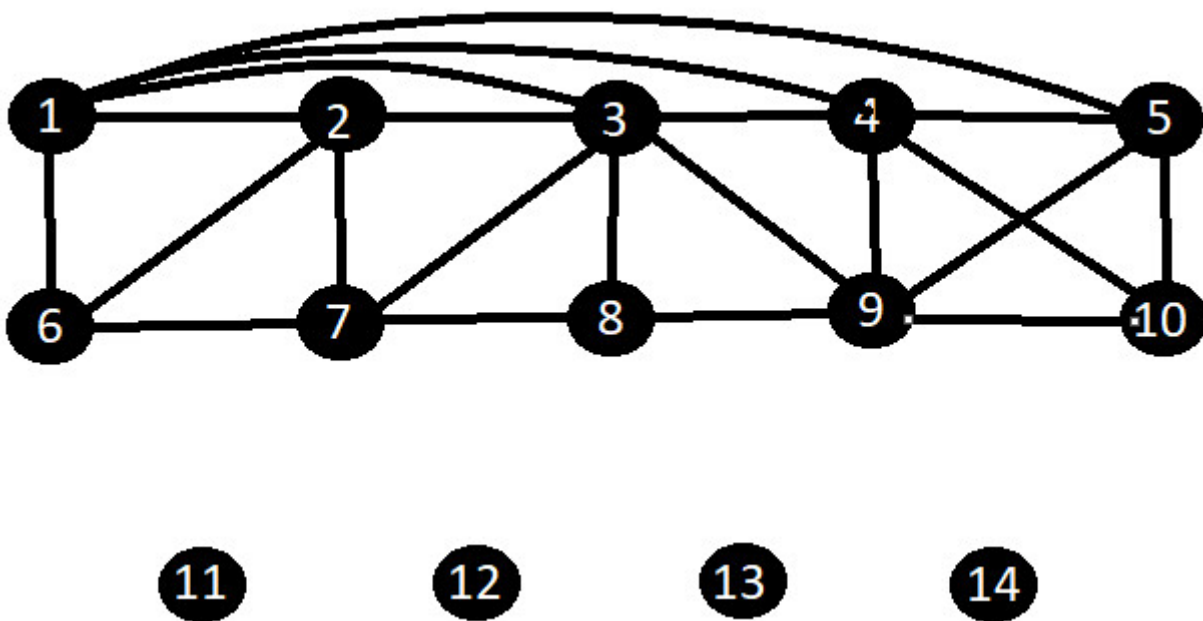
Князев Антон, К-12, Варіант 77, Викладач: Єфремов Микола Сергійович

Перше завдання

```

1 2 3 4 5 6
2 1 3 6 7
3 1 2 7 8 9 4
4 3 9 10 5
5 4 9 10
6 1 2 7
7 6 2 3 8
8 7 3 9
9 8 3 4 5 10
10 9 4 5
11
12
13
14

```

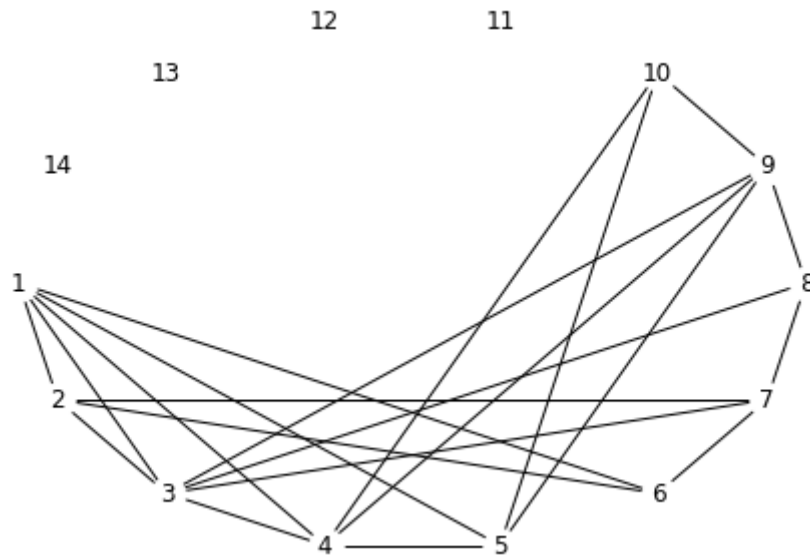


Друге завдання

```

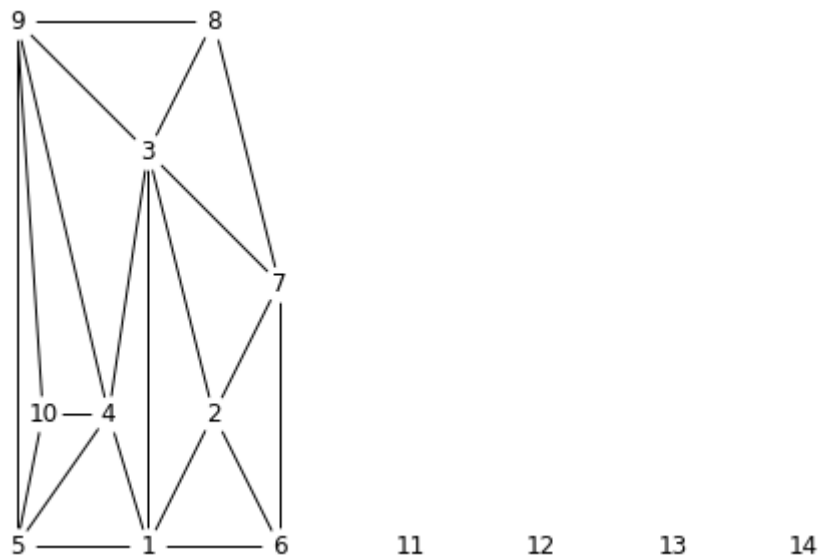
In [47]: import networkx as nx
import numpy as np
import matplotlib.pyplot as plt
g = nx.read_adjlist("graph.txt", nodetype=int)
nx.draw_shell(g, with_labels=True, node_color='white', font_color='black')
plt.show()

```



Третє завдання

```
In [48]: position = { 1 :(-1, 0), 2 :(-0.5, 0.25), 3 :(-1, 0.75), 4 :(-1.3, 0.25), 5 :(-2, 0)
nx.draw(g, pos=position, with_labels=True, node_color='white', font_color='black')
plt.show()
```



Четверте завдання

```
In [49]: params = []
for component in nx.connected_components(g):
    nodedegree = []
    nodeeccentricity = []
    nodediameter = []
    noderadius = []
    sub = g.subgraph(component)
    params.append(nx.number_of_nodes(sub))
    params.append(nx.number_of_edges(sub))
    for u in nx.nodes(sub):
        nodedegree.append(sub.degree(u))
        nodeeccentricity.append(nx.eccentricity(sub, v=u))
```

```

params.append(nodedegree)
params.append(nx.eccentricity(sub))
params.append(min(nodeeccentricity)) #radius
params.append(max(nodeeccentricity)) #diameter
params = np.array(params, dtype=object)
params = params.reshape(5,6)

```

In [50]:

```

i = 0
k = 0
while i < nx.number_connected_components(g):
    j = 0
    print(f"\nComponent № {i+1}\n\
number of nodes:{params[i][0]}\n\
number of edges:{params[i][1]}\n\
each node degree:", end = ' ')
    for el in params[i][2]:
        print(f"'{k+1}':{params[i][2][j]}", ", end = '')
        k+=1
        j+=1
    print("\neach node eccentricity:", end = ' ')
    for key, value in params[i][3].items():
        print(f"'{key}':{value}", ", end = '')
    print(f"\nradius: {params[i][4]}\n\
diameter: {params[i][5]}")
    i+=1

```

```

Component № 1
number of nodes:10
number of edges:21
each node degree: '1':5, '2':4, '3':6, '4':5, '5':4, '6':3, '7':4, '8':3, '9':5, '10':3,
each node eccentricity: '1':2, '2':3, '3':2, '4':2, '5':3, '6':3, '7':3, '8':2, '9':3, '10':3,
radius: 2
diameter: 3

```

```

Component № 2
number of nodes:1
number of edges:0
each node degree: '11':0,
each node eccentricity: '11':0,
radius: 0
diameter: 0

```

```

Component № 3
number of nodes:1
number of edges:0
each node degree: '12':0,
each node eccentricity: '12':0,
radius: 0
diameter: 0

```

```

Component № 4
number of nodes:1
number of edges:0
each node degree: '13':0,
each node eccentricity: '13':0,
radius: 0
diameter: 0

```

```

Component № 5
number of nodes:1
number of edges:0
each node degree: '14':0,
each node eccentricity: '14':0,

```

radius: 0
diameter: 0

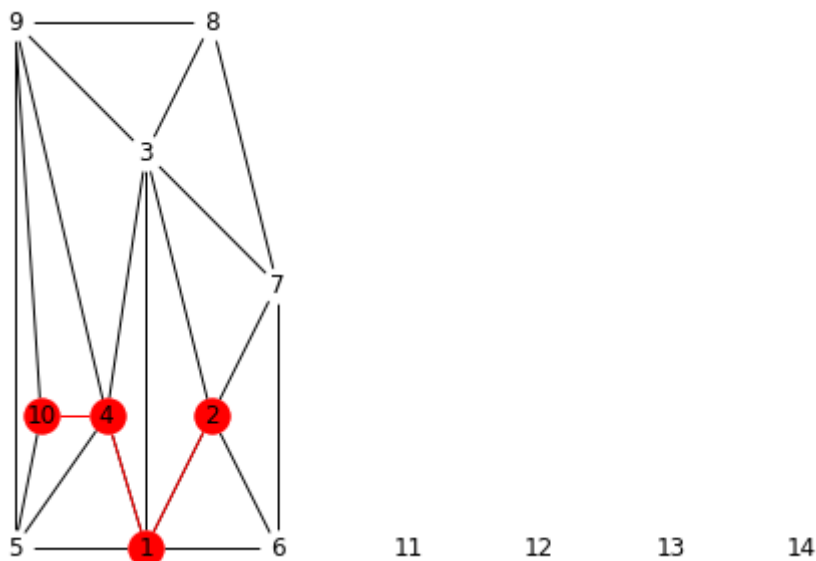
П'яте завдання

In [51]:

```
diameter = []
red_nodes = []
red_edges = []
for component in nx.connected_components(g):
    diam_nodes = []
    diam_vars=[]
    sub = g.subgraph(component)
    ecc= nx.eccentricity(sub)
    diam = max(ecc.values())
    for key, value in ecc.items():
        if value == diam:
            diam_nodes.append(key)
    try:
        i = 0
        while i < len(diam_nodes):
            j = 0
            while j < len(diam_nodes):
                diam_vars.append(nx.shortest_path(sub, source=diam_nodes[i], target=
                    j+=1
                i+=1
            diameter = max(diam_vars, key = len)
            if len(diameter) > 1:
                red_nodes = red_nodes + diameter #add diameter nodes to list of nodes, w
        i = 0
        for el in diameter: #creating list of edges, which should be colored
            red_edges.append((diameter[i], diameter[i+1]))
            i+=1
    except IndexError: #pass, if component is trivial
        pass
```

In [52]:

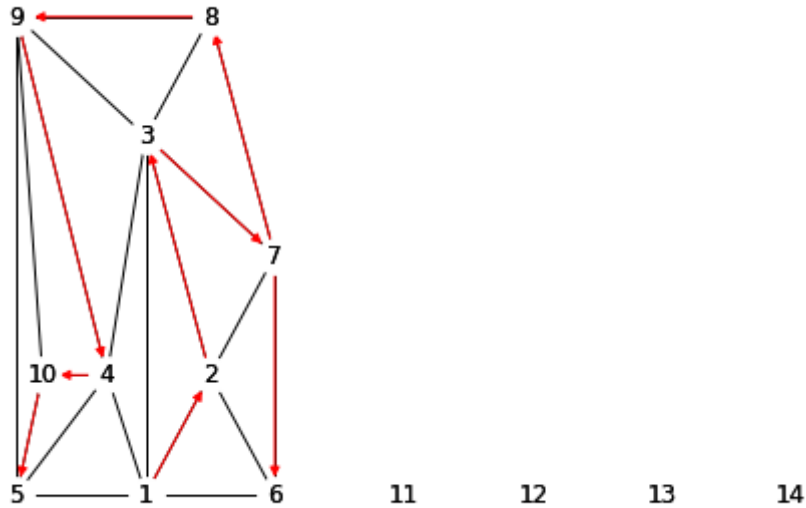
```
nx.draw(g, pos=position, with_labels=True, node_color='white', font_color='black')
nx.draw_networkx_nodes(g, pos=position, nodelist=red_nodes, node_color="red")
nx.draw_networkx_edges(g, pos=position, edgelist=red_edges, edge_color="red",)
plt.draw()
```



Шосте завдання

In [54]:

```
nx.draw(g, pos=position, with_labels=True, node_color='white', font_color='black')
for component in nx.connected_components(g):
    sub = g.subgraph(component)
    nodes = list(component)
    tree = nx.dfs_tree(sub, nodes[0])
    nx.draw(tree, pos=position, with_labels=True, node_color='white', edge_color='r')
plt.draw()
```



In []: