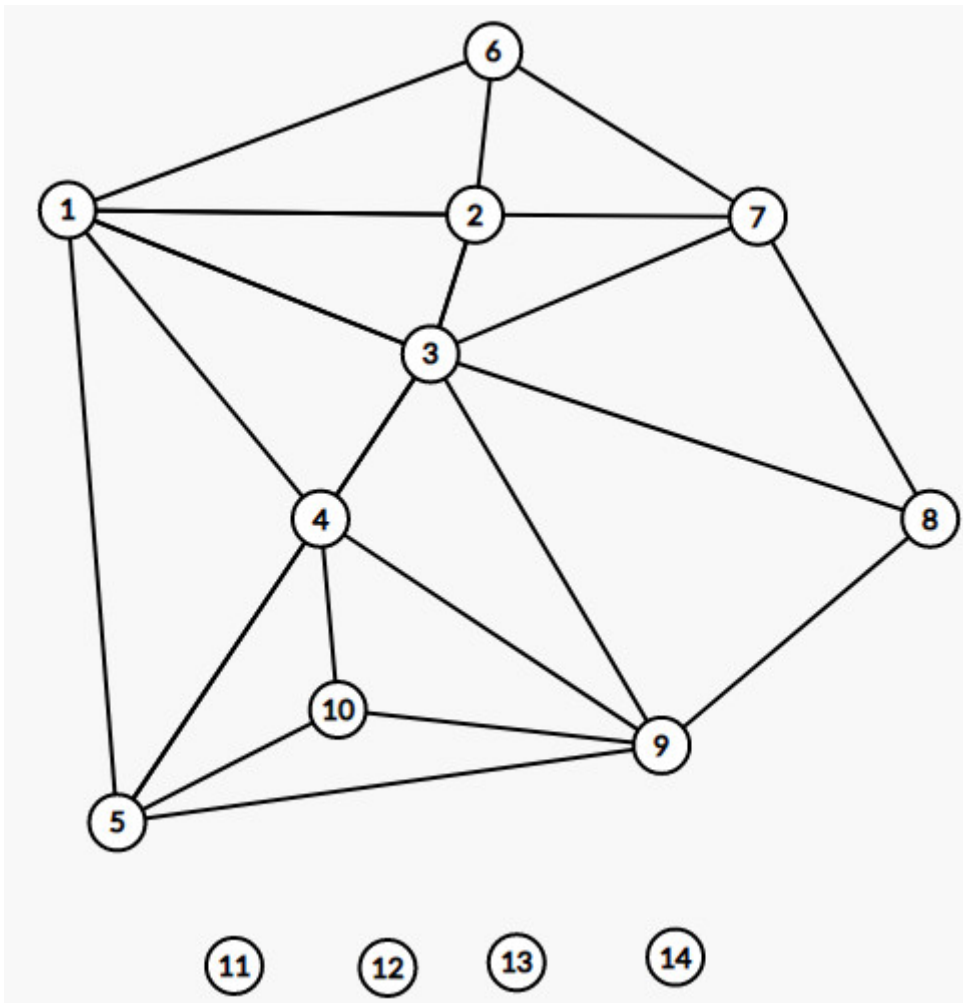Князєв Антон, К-12, Варіант 77, Викладач: Єфремов Микола Сергійович

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
```
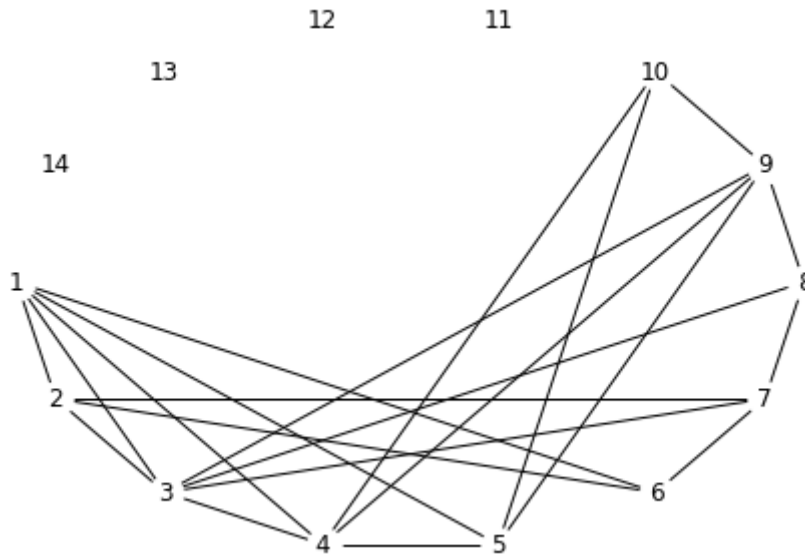

Друге завдання

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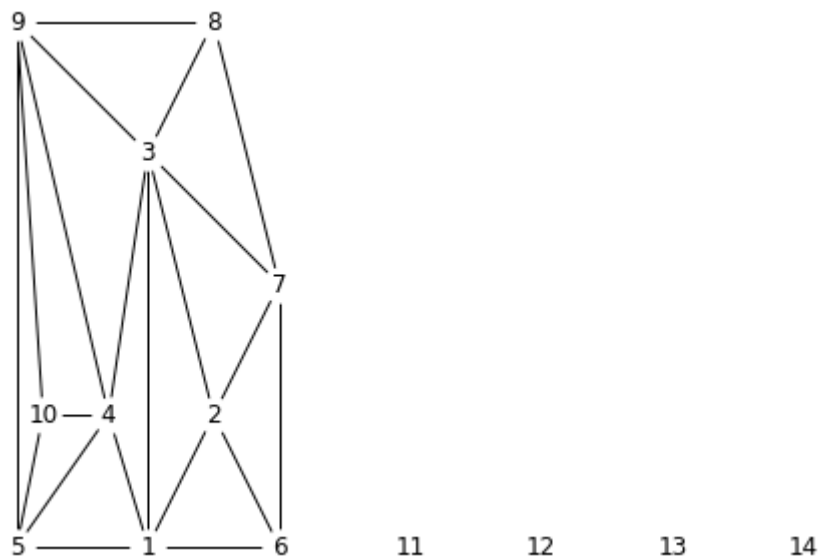
```
nx.draw_shell(g, with_labels=True, node_color='white', font_color='black')
plt.show()
```


Третє завдання

In [112…
```
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 [113…
```
params = []
comps = nx.connected_components(g)
for component in comps:
    nodedegree = []
    nodeeccentricity = []
    nodediameter = []
    noderadius = []
    sub = g.subgraph(component)
    params.append(nx.number_of_nodes(sub))
```

```
        params.append(nx.number_of_edges(sub))
        sub_nodes = nx.nodes(sub)
        for u in sub_nodes:
            nodedegree.append(sub.degree(u))
            nodeeccentricity.append(nx.eccentricity(sub, v=u))
        params.append(nodedegree)
        params.append(nx.eccentricity(sub))
        params.append(min(nodeeccentricity))
        params.append(max(nodeeccentricity))
    params = np.array(params, dtype=object)
    params = params.reshape(5,6)
```

In [114…]
```
i = 0
k = 0
number_of_comps = nx.number_connected_components(g)
while i < number_of_comps:
    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 = ' ')
    list_of_degrees = params[i][2]
    for el in list_of_degrees:
        print(f"'{k+1}':{params[i][2][j]}, ", end = '')
        k+=1
        j+=1
    print("\neach node eccentricity:", end = ' ')
    list_of_ecc = params[i][3].items()
    for key, value in list_of_ecc:
        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, '1
0':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
```
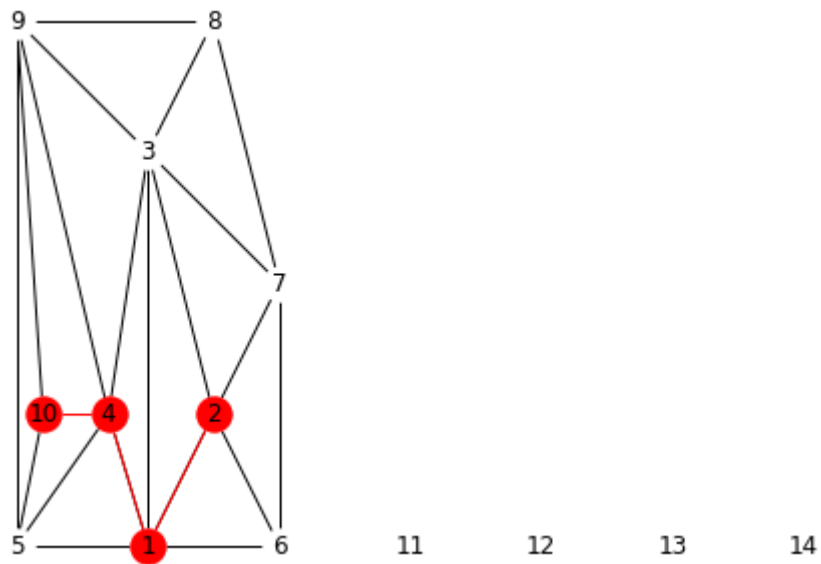
П'яте завдання

In [115…
```python
red_nodes = []
red_edges = []
comps = nx.connected_components(g)
for component in comps:
    diam_nodes = []
    prob_diam=[]
    sub = g.subgraph(component)
    ecc= nx.eccentricity(sub)
    diam = max(ecc.values())
    list_of_ecc = ecc.items()
    for key, value in list_of_ecc:
        if value == diam:
            diam_nodes.append(key)
    len_diam_nodes = len(diam_nodes)
    try:
        i = 0
        while i < len_diam_nodes:
            j = i + 1
            while j < len_diam_nodes:
                len_diam_vars = len(diam_vars)
                prob_diam = nx.shortest_path(sub, source=diam_nodes[i], target=diam_
                len_prob_diam = len(prob_diam)
                if len_prob_diam == diam + 1:
                    break
                j+=1
            if prob_diam:
                break
            i+=1
        len_diameter = len(prob_diam)
        if len_diameter > 1:
            red_nodes = red_nodes + prob_diam
        i = 0
        for el in prob_diam:
            red_edges.append((prob_diam[i], prob_diam[i+1]))
            i+=1
    except IndexError:
        pass
```

In [116…
```python
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 [117…
nx.draw(g, pos=position, with_labels=True, node_color='white', font_color='black')
comps = nx.connected_components(g)
for component in comps:
    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()
```