Exercise 4 - Theory

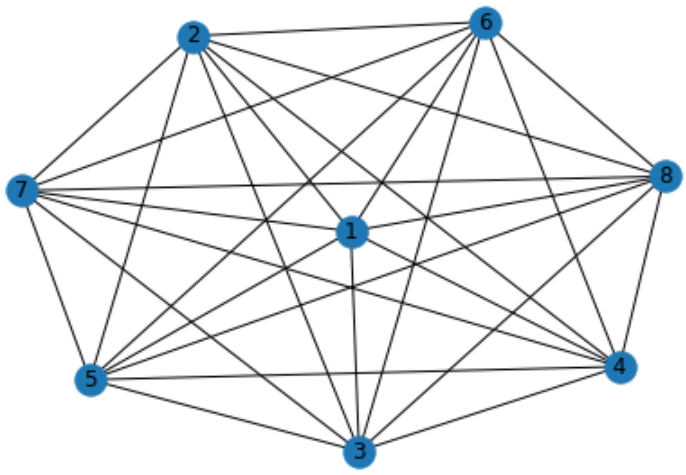
In [ ]:

```
import networkx as nx
```

Problem 1

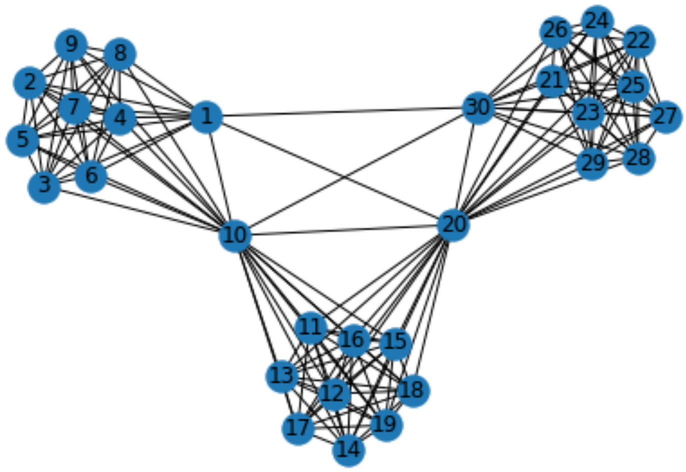
In [ ]:

```
G1 = nx.convert_node_labels_to_integers(nx.complete_graph(8), first_label=1)
nx.draw(G1, with_labels=True)
```



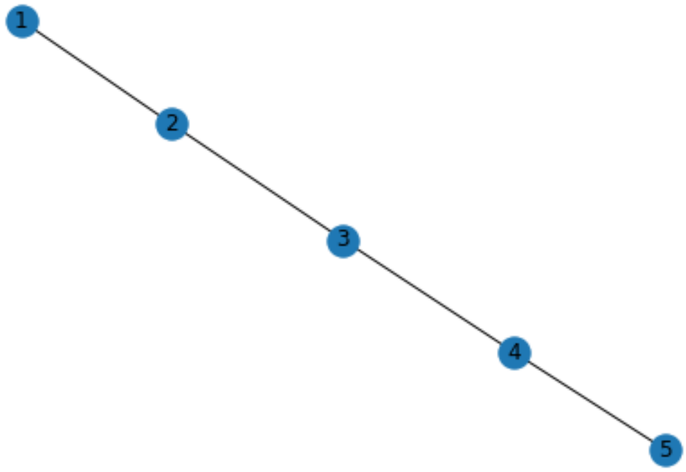
In [ ]:

```
G2_a = nx.convert_node_labels_to_integers(nx.complete_graph(10), first_label=1)
G2_b = nx.convert_node_labels_to_integers(nx.complete_graph(11), first_label=10)
G2_c = nx.convert_node_labels_to_integers(nx.complete_graph(11), first_label=20)
G2 = nx.compose(nx.compose(G2_a, G2_b), G2_c)
G2.add_edges_from([(1,30), (1,20), (10,20), (10,30), (20,30)])
nx.draw(G2, with_labels=True)
```



In [ ]:

```
G3 = nx.Graph()
G3.add_edges_from([(1,2), (2,3), (3,4), (4,5)])
nx.draw(G3, with_labels=True)
```



a) Order the three graphs by their diameter without computing.

Since G1 is complete its diameter is 1. In G2 each cluster is connected to an outgoing node and that one either to a hub or to an outgoing node of another cluster. Thus the longest shortest path is 3. G3 forms just a single path of length 4. In descending order we have: G3, G2, G1.

b) Order the three graphs by their density without computing.

Again since G1 is complete the density is 1. The density of both G2 and G3 is low since there are a lot of possible edges missing but since G2 has 30 nodes instead of 5 the number of possible edges is much larger than that of G3 and so its density is probably also a little lower than that of G3. Thus G1, G3, G2 in descending order.

c) Order the three graphs by their average clustering coefficient without computing.

The average clustering coefficient of G1 is 1 since every node is connected. That of G3 is 0 because no node has any connected neighbors. Finally G2 lies strictly inbetween. Thus we have G1, G2, G3 in descending order.

Problem 2

Suppose the following table contains measurements about 2 real world networks and one ER reference network. The first column gives the number of nodes, the second the number of edges and the third refers to the average clustering coefficient:

Graph	n	m	c
X1	4941	6594	0.08
X2	125	560	0.07
X3	256985	7778954	0.09

Unfortunately, the labels are lost and therefore also the identification of the random network among them. Can you deduce from the measurements which network is most likely to be the random one?

Heuristically speaking if we look at a given node in an ER graph $G(n,p)$ we would expect there to be $\frac{k(k-1)}{2}p$ edges between its neighbors where k is its degree. To calculate its clustering coefficient we divide by the number of possible edges $\frac{k(k-1)}{2}$ and thus we are left with just p as the clustering coefficient of a random node. Therefore the average clustering coefficient of the ER graph has to be $c = p$.

Let's now suppose that each graph from above is an ER graph $G(n,p)$ where $p = \frac{m}{\binom{n}{2}}$ is the probability at which we would expect the random model to have m edges. Now we only need to compare for which graph the clustering coefficient c matches the probability p.

$$p_{X1} = \frac{1099}{2034045} \approx 0.00054$$
$$p_{X2} = \frac{56}{775} \approx 0.07225$$
$$p_{X3} = \frac{3889477}{16510258310} \approx 0.00023$$

From this we conclude that $X2$ must be the ER model since its average clustering coefficient matches its probability