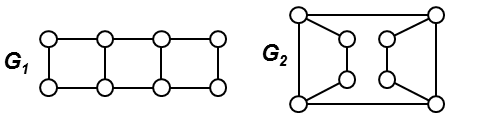
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a) | b) | c) | d) | e) |
| f) | g) | h) | i) | j) |

1. Decide, which pairs of the following graphs are isomprphic and which are not.



2. How many isomorphisms are there between G1 a G2?

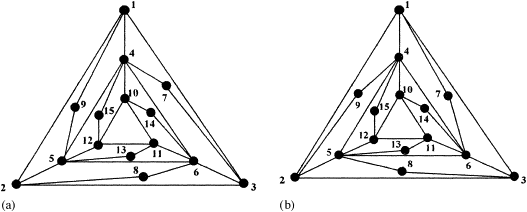
3. In the picture bellow, what is the number of such bijections between the nodes of graph G1 and G2 which are not isomorphisms?

|  |  |
| --- | --- |
| a) | b) |
| c) | d) |

4. There are two undirected graphs, both have *n* nodes and the degree sequence of both graphs is

(*n*─1, *n*─2, *n*─3, *n*─4, ..., *n*/2+1, *n*/2, *n*/2, *n*/2─1, *n*/2─2, ..., 3, 2, 1). As you can see, nearly all node degrees are different with the exception of two nodes which share the same degree *n*/2. The fact that node degrees are different may significantly speed up the process of deciding whether two graphs are isomorphic. What is the asymptotic complexity of the process you can think of in this case?

5. Decide whether the two given graphs are isomorphic. Describe the decision process, remember, it shoud be effective.



**Trees**

\* 1. Create the certificates of the trees in cases a) - d)

|  |  |  |  |
| --- | --- | --- | --- |
| a) | b) | c) | d) |

**\*** 2**.** Reconstruct the tree from the given certificate.

a) 0011

b) 000101100101011

c) 00010110010110010111

d) 0000110001011110010001011111

\* 3. We know the certificate of a particular tree. Describe how to find the number of leaves of the tree using only the certificate without reconstructing the tree itself.

4. We know the certificate of a particular tree. Describe how to find the maximum degree of all nodes in the tree using only the certificate without reconstructing the tree itself.

5. A tree is said to be of type T(1,3) when its node degrees are only 1 or 3. Describe informally how the certificate of such tree looks like and design an algorithm based on the tree certificate which will decide whether the corresponding tree is of type T(1,3).