Exercise 1:
Let P_1 and P_2 be two permutation matrices. Is $P_1 \times P_2$ also a permutation matrix? Argue for or against your answer.

Exercise 2:

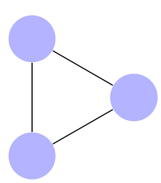
1. Draw the graphs G_A and G_B for which the following 2 adjacency matrices A and B are given.

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} B = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

2. Are the two graphs isomorphic?
3. How many different representations (in terms of adjacency matrices) of G_A are there?
4. How many different representations (in terms of adjacency matrices) of G_B are there?
5. Is there a permutation matrix P such that $A = P\,(PB)^T$ holds?
6. If so, give all matrices P, such that $A = P\,(PB)^T$ holds.

Exercise 3:

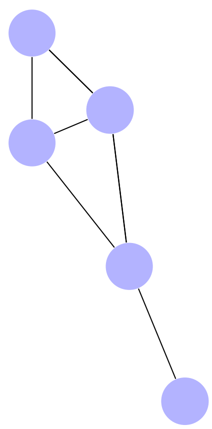
Given the following graph:



1. Give an adjacency matrix A for the graph. (How many different are there?)
2. For your chosen adjacency matrix, how many permutation matrices P are there, such that $A = P\,(PA)^T$ holds? (Remark: this number corresponds to the size of the so-called “automorphism group” of the graph).

Exercise 4:

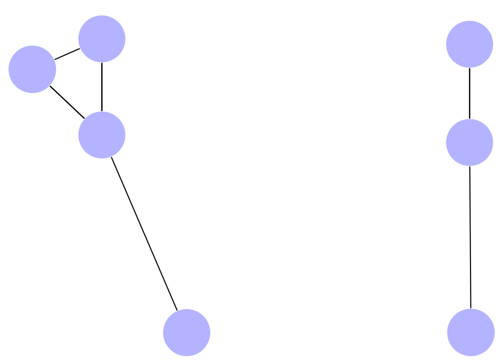
Given the following graph:



1. Give an adjacency matrix A for the graph.
2. For your chosen adjacency matrix, how many permutation matrices P are there, such that $A = P\,(PA)^T$ holds?

Exercise 5*:

Given the following two graphs G_A (left) and G_B (right):



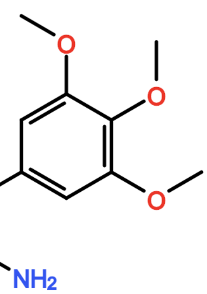
1. Give adjacency matrices for G_A and G_B.
2. Is G_B a subgraph of G_A?
3. How many different ways are there to find G_B as a subgraph in G_A? (i.e., assuming as adjacency matrix A and B for graphs G_A and G_B, how many leaf-nodes would the search the of the Ullmann algorithm have?)

4. How many different ways are there to find G_B as an induced subgraph in G_A?

Exercise 6:
The following is from the unit-testing of the graph theory assignment. Explain the expected result 10.

```
>>> A = np.array([[ 0,  1,  0,  0,  1],  \
                  [ 1,  0,  1,  0,  0], \
                  [ 0,  1,  0,  1,  0], \
                  [ 0,  0,  1,  0,  1], \
                  [ 1,  0,  0,  1,  0]])
>>> numIsomorphisms(A, A)
10
```

Exercise 7*:
Use sigma aldrich `https://www.sigmaaldrich.com/catalog/search/substructure/SubstructureSearchPage`to look for chemical structures. How many structures can you find which have the following as a substructure?



Can you find the price for the compounds you found? Any guesses why not?