

```

In [2]: ### Create adjacency matrices
# We want to create the adjacency matrices efficiently now

#Create all possible binary strings with the length of the number of elem
#size of matrix
import itertools
import numpy as np

def all_adj_matrix(n):
    sl = 1/2 * (n-1)**2 + 1/2*(n-1) # string length
    allstrings = []
    strings = itertools.product([0, 1], repeat=int(sl))

    for item in strings:
        allstrings.append(item)

    p = len(allstrings) # number of matrices we need
    adjmat = [] # list of adjmats

    for k in range(p):
        v = np.array(allstrings[k])
        A = np.zeros((n, n))

        # Append the strings in the upper part of the matrix
        idx = 0
        for i in range(n):
            for j in range(i+1, n):
                if idx < len(v):
                    A[i][j] = v[idx]
                    idx += 1

        # Reflect about the diagonal
        for i in range(n):
            for j in range(i+1, n):
                A[j][i] = A[i][j]
        A = A.tolist() #new line here to produce lists of lists and not n
        adjmat.append(A)
    return adjmat

```

In []: