**Problem #2965: Find Missing and Repeated Values (Easy)**

<https://leetcode.com/problems/find-missing-and-repeated-values/description/>

**My Solution:**

1. Let n be the length of grid. So number of rows of the square matrix grid is n and number of columns is n.
2. Initialize mydict to be an empty dictionary.
3. For I in range of n and for j in range of n (inner loop), if grid with row I and column j in not in mydict keys, then put it in mydict as the key with value 1. However, if it is already in mydict, then the value of the key for the grid element is incremented by 1.

Also, grid value is assigned to repeated.

1. The missing number is the set difference of set of range from 1 to ((n\*\*2 )+ 1) and set of mydict keys.
2. Return a list with repeated as the first element and missing converted to a list and element of this list at index 0.

class Solution:

def findMissingAndRepeatedValues(self, grid: List[List[int]]) -> List[int]:

n = len(grid)

mydict = {}

for i in range(n):

for j in range(n):

if grid[i][j] not in mydict.keys():

mydict[grid[i][j]] = 1

else:

mydict[grid[i][j]] += 1

repeated = grid[i][j]

missing = set(range(1, (n \*\* 2) + 1)) - set(mydict.keys())

return[repeated, list(missing)[0]]