Problem #566: Reshape the matrix (Easy)

----------------------------------------------------------

<https://leetcode.com/problems/reshape-the-matrix/>

My Solution:

1) Flatten the given matrix to a 1-dimensional array.

2) If the number of elements in the new matrix is not the same as in the given matrix,

return the given matrix.

3) Iterate through the original matrix and put the results in res\_mat with the new dimensions r and c.

4) Return the result matrix res\_mat

Runtime beats 94.22%

class Solution:

def matrixReshape(self, mat: List[List[int]], r: int, c: int) -> List[List[int]]:

# Copy the elements of mat into a 1-dimensional array

arr = [] # initialize the 1-dimensional array

m = len(mat) # number of rows

n = len(mat[0]) # number of columns

for i in range(m):

for j in range(n):

arr.append(mat[i][j])

p = len(arr)

if p != c \* r: # total number of elements in mat is not the same as new matrix

return mat

res\_mat = [] # initialize the result matrix

temp\_arr = [] # initialize a temp array

for i in range(r):

if i:

res\_mat.append(temp\_arr) # a row is complete and so copy the row to res\_mat

temp\_arr = [] # initialize temp\_arr to start a new row

for j in range(c):

temp\_arr.append(arr[i\*c + j]) #append the elements onto the new matrix

res\_mat.append(temp\_arr) # The last row of the new matrix is appended to it

return res\_mat