Spiral Matrix

Question: Given a matrix of m x n elements (m rows, n columns), return all elements of the matrix in spiral order.

For example:

Given the following matrix:

[[1, 2, 3],

[4, 5, 6],

[7, 8, 9]]

You should return [1,2,3,6,9,8,7,4,5].

Solutions:

```
class Solution:
    # @param matrix, a list of lists of integers
    # @return a list of integers
    def spiralOrder(self, matrix):
        if matrix == []: return []
        res = []
        maxUp = maxLeft = 0
        maxDown = len(matrix) - 1
        maxRight = len(matrix[0]) - 1
        direct = 0 # 0 go right, 1 go down, 2 go left, 3 go up
        while True:
            if direct == 0: # go right
                for i in range(maxLeft, maxRight + 1): res.append(matrix[maxUp][i])
                maxUp += 1
            elif direct == 1: # go down
                for i in range(maxUp, maxDown + 1): res.append(matrix[i][maxRight])
```

```
            maxRight -= 1
        elif direct == 2: # go left
            for i in reversed(range(maxLeft, maxRight + 1)):
res.append(matrix[maxDown][i])
            maxDown -= 1
        else: # go up
            for i in reversed(range(maxUp, maxDown + 1)):
res.append(matrix[i][maxLeft])
            maxLeft += 1
        if maxUp > maxDown or maxLeft > maxRight: return res
        direct = (direct + 1) % 4


Solution().spiralOrder([ [ 1, 2, 3 ],
                         [ 4, 5, 6 ],
                         [ 7, 8, 9 ] ])
```