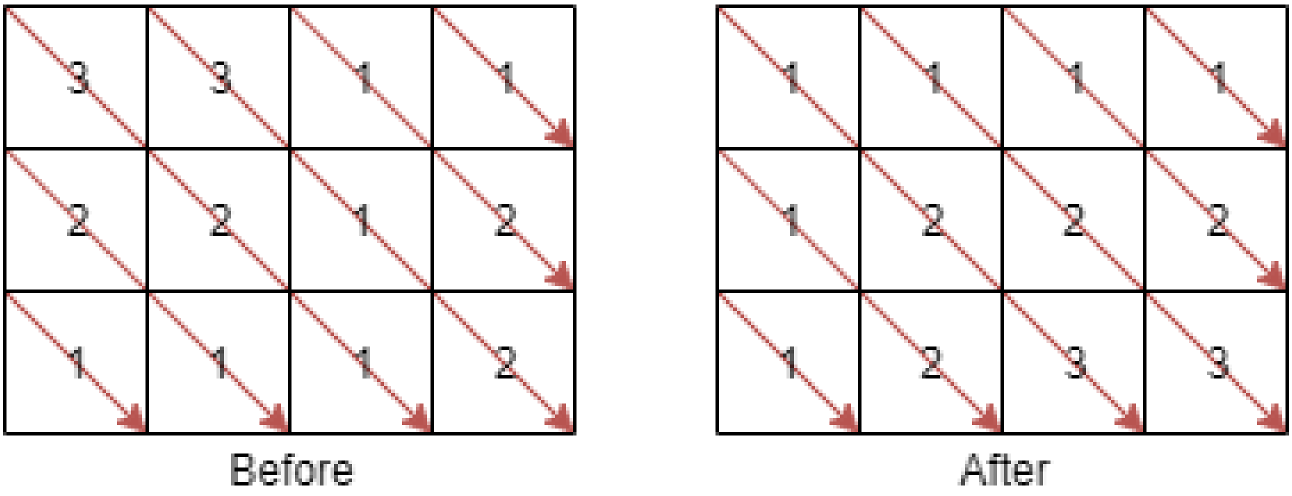
1329. Sort the Matrix Diagonally

A **matrix diagonal** is a diagonal line of cells starting from some cell in either the topmost row or leftmost column and going in the bottom-right direction until reaching the matrix's end. For example, the **matrix diagonal** starting from `mat[2][0]`, where `mat` is a `6 x 3` matrix, includes cells `mat[2][0]`, `mat[3][1]`, and `mat[4][2]`.

Given an `m x n` matrix `mat` of integers, sort each **matrix diagonal** in ascending order and return *the resulting matrix*.



Input: mat = [[3,3,1,1],[2,2,1,2],[1,1,1,2]]
Output: [[1,1,1,1],[1,2,2,2],[1,2,3,3]]

Example 2:

Input: mat = [[11,25,66,1,69,7],[23,55,17,45,15,52],[75,31,36,44,58,8],[22,27,33,25,68,4],[84,28,14,11,5,50]]
Output: [[5,17,4,1,52,7],[11,11,25,45,8,69],[14,23,25,44,58,15],[22,27,31,36,50,66],[84,28,75,33,55,68]]

```
def diagonalSort(self, mat: List[List[int]]) -> List[List[int]]:
        diagonals = collections.defaultdict(list)
        for i in range(len(mat)):
            for j in range(len(mat[0])):
                if i-j in diagonals:
                    diagonals[i-j].append(mat[i][j])
                else:
                    diagonals[i-j] = [mat[i][j]]
        for ele in diagonals:
            diagonals[ele] = sorted(diagonals[ele],reverse=True)
        for i in range(len(mat)):
            for j in range(len(mat[0])):
                temp = diagonals[i-j].pop()
                mat[i][j] = temp
        return mat
```