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.

Example 1:

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]]

Program:

```
def diagonalSort(mat):
    from collections import defaultdict
    import heapq
    m, n = len(mat), len(mat[0])
    diagonals = defaultdict(list)
    for i in range(m):
        for j in range(n):
            diagonals[i - j].append(mat[i][j])
    for key in diagonals:
        diagonals[key].sort()
```

```
    for i in range(m):
        for j in range(n):
            mat[i][j] = diagonals[i - j].pop(0)
    return mat
mat = [
    [3, 3, 1, 1],
    [2, 2, 1, 2],
    [1, 1, 1, 2]
]
print(diagonalSort(mat))
```

Output:

Time complexity:

O(m.nlogmin(m,n))