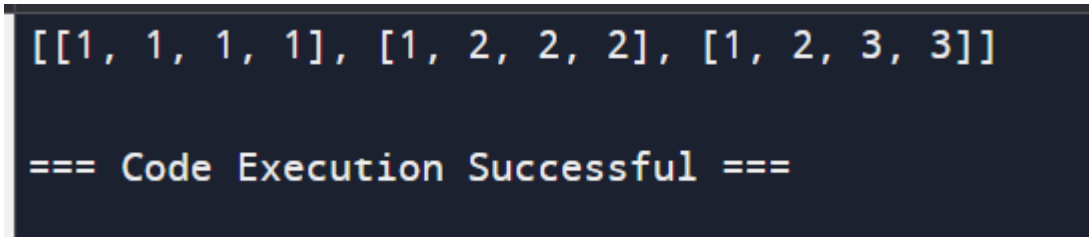
54. Sort the Matrix Diagonally A matrix diagonal is a diagonal line of cells starting from some cell in either the topmost row or leftmost column and going in the bottom-right direction until reaching the matrix's end. For example, the matrix diagonal starting from mat[2][0], where mat is a 6 x 3 matrix, 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 1Input: 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):
    m, n = len(mat), len(mat[0])

    # Helper function to traverse and sort each diagonal
    def traverse_and_sort(i, j):
        diagonal = []
        while i < m and j < n:
            diagonal.append(mat[i][j])
            i += 1
            j += 1
        diagonal.sort()
        i, j = i - 1, j - 1
        while i >= 0 and j >= 0:
            mat[i][j] = diagonal.pop()
            i -= 1
            j -= 1

    # Traverse diagonals starting from the top row
    for j in range(n):
        traverse_and_sort(0, j)

    # Traverse diagonals starting from the leftmost column
    for i in range(1, m):
        traverse_and_sort(i, 0)

    return mat

# Example usage:
mat = [[3, 3, 1, 1], [2, 2, 1, 2], [1, 1, 1, 2]]
print(diagonalSort(mat))  # Output: [[1, 1, 1, 1], [1, 2, 2, 2], [1, 2, 3, 3]]
```

OUTPUT:-



TIME COMPLEXITY:-O((m+n) log(m+n))