1030. Matrix Cells in Distance Order

You are given four integers `row`, `cols`, `rCenter`, and `cCenter`. There is a `rows x cols` matrix and you are on the cell with the coordinates `(rCenter, cCenter)`.

Return *the coordinates of all cells in the matrix, sorted by their **distance** from `(rCenter, cCenter)` from the smallest distance to the largest distance*. You may return the answer in **any order** that satisfies this condition.

The **distance** between two cells `(r<sub>1</sub>, c<sub>1</sub>)` and `(r<sub>2</sub>, c<sub>2</sub>)` is `|r<sub>1</sub> - r<sub>2</sub>| + |c<sub>1</sub> - c<sub>2</sub>|`.

Example 1:

```
Input: rows = 1, cols = 2, rCenter = 0, cCenter = 0
Output: [[0,0],[0,1]]
Explanation: The distances from (0, 0) to other cells are: [0,1]
```

Example 2:

```
Input: rows = 2, cols = 2, rCenter = 0, cCenter = 1
Output: [[0,1],[0,0],[1,1],[1,0]]
Explanation: The distances from (0, 1) to other cells are: [0,1,1,2]
The answer [[0,1],[1,1],[0,0],[1,0]] would also be accepted as correct.
```

Example 3:

```
Input: rows = 2, cols = 3, rCenter = 1, cCenter = 2
Output: [[1,2],[0,2],[1,1],[0,1],[1,0],[0,0]]
Explanation: The distances from (1, 2) to other cells are: [0,1,1,2,2,3]
There are other answers that would also be accepted as correct, such as
[[1,2],[1,1],[0,2],[1,0],[0,1],[0,0]].
```

Constraints:

- `1 <= rows, cols <= 100`
- `0 <= rCenter < rows`
- `0 <= cCenter < cols`

```
def allCellsDistOrder(self, R: int, C: int, r0: int, c0: int) ->
List[List[int]]:
        res = [[x,y] for x in range(R) for y in range(C)]
```

```
        res.sort(key=lambda x: abs(x[0]-r0) + abs(x[1]-c0))
        return res
```