1030. Matrix Cells in Distance Order

We are given a matrix with `R` rows and `C` columns has cells with integer coordinates `(r, c)`, where `0 <= r < R` and `0 <= c < C`.

Additionally, we are given a cell in that matrix with coordinates `(r0, c0)`.

Return the coordinates of all cells in the matrix, sorted by their distance from `(r0, c0)` from smallest distance to largest distance. Here, the distance between two cells `(r1, c1)` and `(r2, c2)` is the Manhattan distance, `|r1 - r2| + |c1 - c2|`. (You may return the answer in any order that satisfies this condition.)

Example 1:

Input: R = 1, C = 2, r0 = 0, c0 = 0
Output: [[0,0],[0,1]]
Explanation: The distances from (r0, c0) to other cells are: [0,1]

Example 2:

Input: R = 2, C = 2, r0 = 0, c0 = 1
Output: [[0,1],[0,0],[1,1],[1,0]] **Explanation:** The distances from (r0, c0) 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: R = 2, C = 3, r0 = 1, c0 = 2
Output: [[1,2],[0,2],[1,1],[0,1],[1,0],[0,0]]
Explanation: The distances from (r0, c0) 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]].

```
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
```