Solution 1   Solution 2   Solution 3

```python
# Copyright © 2020 AlgoExpert, LLC. All rights reserved.

# O(n^2) time | O(n) space - where n is the number of coordinates
def rectangleMania(coords):
    coordsTable = getCoordsTable(coords)
    return getRectangleCount(coords, coordsTable)


def getCoordsTable(coords):
    coordsTable = {"x": {}, "y": {}}
    for coord in coords:
        x, y = coord
        if x not in coordsTable["x"]:
            coordsTable["x"][x] = []
        coordsTable["x"][x].append(coord)
        if y not in coordsTable["y"]:
            coordsTable["y"][y] = []
        coordsTable["y"][y].append(coord)
    return coordsTable


def getRectangleCount(coords, coordsTable):
    rectangleCount = 0
    for coord in coords:
        lowerLeftY = coord[1]
        rectangleCount += clockwiseCountRectangles(coord, coordsTable, UP, lowerLeftY)
    return rectangleCount


def clockwiseCountRectangles(coord1, coordsTable, direction, lowerLeftY):
    x1, y1 = coord1
    if direction == DOWN:
        relevantCoords = coordsTable["x"][x1]
        for coord2 in relevantCoords:
            lowerRightY = coord2[1]
            if lowerRightY == lowerLeftY:
                return 1
        return 0
    else:
        rectangleCount = 0
        if direction == UP:
            relevantCoords = coordsTable["x"][x1]
            for coord2 in relevantCoords:
                y2 = coord2[1]
                isAbove = y2 > y1
                if isAbove:
                    rectangleCount += clockwiseCountRectangles(coord2, coordsTable, RIGHT, lowerLeftY)
        elif direction == RIGHT:
            relevantCoords = coordsTable["y"][y1]
            for coord2 in relevantCoords:
                x2 = coord2[0]
                isRight = x2 > x1
                if isRight:
                    rectangleCount += clockwiseCountRectangles(coord2, coordsTable, DOWN, lowerLeftY)
        return rectangleCount


UP = "up"
RIGHT = "right"
DOWN = "down"
```