```swift
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

class Program {
    let UP = "up"
    let DOWN = "down"
    let LEFT = "left"
    let RIGHT = "right"

    func coordToString(_ coord: [Int]) -> String {
        let x = coord[0]
        let y = coord[1]

        return "\(x)-\(y)"
    }

    // O(n^2) time | O(n^2) space
    func rectangleMania(_ coords: [[Int]]) -> Int {
        let coordsTable = getCoordsTable(coords)
        return getRectangleCount(coords, coordsTable)
    }

    func getCoordsTable(_ coords: [[Int]]) -> [String: [String: [[Int]]]] {
        var coordsTable = [String: [String: [[Int]]]]()

        for coord1 in coords {
            var coord1Directions: [String: [[Int]]] = [UP: [], DOWN: [], LEFT: [], RIGHT: []]

            for coord2 in coords {
                let coord2Direction = getCoordDirection(coord1, coord2)

                if var coordinatesForDirection = coord1Directions[coord2Direction] {
                    coordinatesForDirection.append(coord2)
                    coord1Directions[coord2Direction] = coordinatesForDirection
                }
            }

            let coords1String = coordToString(coord1)
            coordsTable[coords1String] = coord1Directions
        }

        return coordsTable
    }

    func getCoordDirection(_ coord1: [Int], _ coord2: [Int]) -> String {
        let x1 = coord1[0]
        let y1 = coord1[1]

        let x2 = coord2[0]
        let y2 = coord2[1]

        if y1 == y2 {
            if x1 < x2 {
                return RIGHT
            } else {
                return LEFT
            }
        } else if x1 == x2 {
            if y1 < y2 {
                return UP
            } else {
                return DOWN
            }
        }

        return ""
    }

    func getRectangleCount(_ coords: [[Int]], _ coordsTable: [String: [String: [[Int]]]]) -> Int {
        var rectangleCount = 0

        for coord in coords {
            rectangleCount += clockwiseCountRectangles(coord, coordsTable, UP, coord)
        }

        return rectangleCount
    }

    func clockwiseCountRectangles(_ coord: [Int], _ coordsTable: [String: [String: [[Int]]]], _ direction: String, _ origin: [Int]) -> Int {
        let coordString = coordToString(coord)

        if direction == LEFT {
            if let directionsForCoordinate = coordsTable[coordString], let coordinatesForDirection = directionsForCoordinate[direction], coordinatesForDirection.contains(origin) {
                return 1
            } else {
                return 0
            }
        } else {
            var rectangleCount = 0
            let nextDirection = getNextClockwiseDirection(direction)

            if let directionsForCoordinate = coordsTable[coordString], let coordinatesForDirection = directionsForCoordinate[direction] {
                for nextCoord in coordinatesForDirection {
                    rectangleCount += clockwiseCountRectangles(nextCoord, coordsTable, nextDirection, origin)
                }
            }

            return rectangleCount
        }
    }

    func getNextClockwiseDirection(_ direction: String) -> String {
        if direction == UP {
            return RIGHT
        }

        if direction == RIGHT {
            return DOWN
        }

        if direction == DOWN {
            return LEFT
        }

        return ""
    }
```

```
}

```