

Solution 1Solution 2Solution 3

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 package main
4
5 type Coord struct {
6     X, Y int
7 }
8
9 type Direction int
10
11 const (
12     None Direction = iota - 1
13     Up
14     Down
15     Left
16     Right
17 )
18
19 // O(n^2) time | O(n) space - where n is the number of coordinates
20 func RectangleMania(coords []Coord) int {
21     coordsTable := getCoordsTable(coords)
22     return getRectangleCount(coords, coordsTable)
23 }
24
25 type CoordSet map[Coord]struct{}
26 type CoordsTable struct {
27     Xs, Ys map[int]CoordSet
28 }
29
30 func getCoordsTable(coords []Coord) CoordsTable {
31     table := CoordsTable{
32         Xs: map[int]CoordSet{},
33         Ys: map[int]CoordSet{},
34     }
35     for _, coord := range coords {
36         x, y := coord.X, coord.Y
37         if _, found := table.Xs[x]; !found {
38             table.Xs[x] = CoordSet{}
39         }
40         table.Xs[x][coord] = struct{}{}
41         if _, found := table.Ys[y]; !found {
42             table.Ys[y] = CoordSet{}
43         }
44         table.Ys[y][coord] = struct{}{}
45     }
46     return table
47 }
48
49 func getRectangleCount(coords []Coord, coordsTable CoordsTable) int {
50     count := 0
51     for _, coord := range coords {
52         lowerLeftY := coord.Y
53         count += clockwiseCountRectangles(coord, coordsTable, Up, lowerLeftY)
54     }
55     return count
56 }
57
58 func clockwiseCountRectangles(coord Coord, coordsTable CoordsTable, direction Direction, lowerLeftY int) int {
59     if direction == Down {
60         relevantCoords := coordsTable.Xs[coord.X]
61         for coord2 := range relevantCoords {
62             lowerRightY := coord2.Y
63             if lowerRightY == lowerLeftY {
64                 return 1
65             }
66         }
67         return 0
68     }
69
70     if direction == Up {
71         rectangleCount := 0
72         relevantCoords := coordsTable.Xs[coord.X]
73         for coord2 := range relevantCoords {
74             if coord2.Y > coord.Y {
75                 rectangleCount += clockwiseCountRectangles(coord2, coordsTable, Right, lowerLeftY)
76             }
77         }
78         return rectangleCount
79     }
80
81     if direction == Right {
82         rectangleCount := 0
83         relevantCoords := coordsTable.Ys[coord.Y]
84         for coord2 := range relevantCoords {
85             if coord2.X > coord.X {
86                 rectangleCount += clockwiseCountRectangles(coord2, coordsTable, Down, lowerLeftY)
87             }
88         }
89         return rectangleCount
90     }
91     return 0
92 }
93
```

