

Solution 1Solution 2Solution 3

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 using System.Collections.Generic;
4
5 public class Program {
6     static string UP = "up";
7     static string RIGHT = "right";
8     static string DOWN = "down";
9
10
11 // O(n^2) time | O(n) space - where n is the number of coordinates
12 public static int RectangleMania(Point[] coords) {
13     Dictionary<string, Dictionary<int, List<Point> > > coordsTable = getCoordsTable(
14         coords);
15     return getRectangleCount(coords, coordsTable);
16 }
17
18 public static Dictionary<string, Dictionary<int, List<Point> > > getCoordsTable(
19     Point[] coords) {
20     Dictionary<string, Dictionary<int,
21         List<Point> > > coordsTable = new Dictionary<string,
22         Dictionary<int,
23         List<Point> > >();
24     coordsTable.Add("x", new Dictionary<int, List<Point> >());
25     coordsTable.Add("y", new Dictionary<int, List<Point> >());
26     foreach (Point coord in coords) {
27         if (!coordsTable["x"].ContainsKey(coord.x)) {
28             coordsTable["x"].Add(coord.x, new List<Point>());
29         }
30         if (!coordsTable["y"].ContainsKey(coord.y)) {
31             coordsTable["y"].Add(coord.y, new List<Point>());
32         }
33         coordsTable["x"][coord.x].Add(coord);
34         coordsTable["y"][coord.y].Add(coord);
35     }
36     return coordsTable;
37 }
38
39 public static int getRectangleCount(Point[] coords, Dictionary<string, Dictionary<int,
40     List<Point> > > coordsTable)
41 {
42     int rectangleCount = 0;
43     foreach (Point coord in coords) {
44         int lowerLeftY = coord.y;
45         rectangleCount += clockwiseCountRectangles(coord, coordsTable, UP,
46             lowerLeftY);
47     }
48     return rectangleCount;
49 }
50
51 public static int clockwiseCountRectangles(
52     Point coord1,
53     Dictionary<string, Dictionary<int, List<Point> > > coordsTable,
54     string direction,
55     int lowerLeftY
56 ) {
57     if (direction == DOWN) {
58         List<Point> relevantCoords = coordsTable["x"][coord1.x];
59         foreach (Point coord2 in relevantCoords) {
60             int lowerRightY = coord2.y;
61             if (lowerRightY == lowerLeftY) return 1;
62         }
63         return 0;
64     } else {
65         int rectangleCount = 0;
66         if (direction == UP) {
67             List<Point> relevantCoords = coordsTable["x"][coord1.x];
68             foreach (Point coord2 in relevantCoords) {
69                 bool isAbove = coord2.y > coord1.y;
70                 if (isAbove) rectangleCount += clockwiseCountRectangles(
71                     coord2, coordsTable, RIGHT, lowerLeftY);
72             }
73         } else if (direction == RIGHT) {
74             List<Point> relevantCoords = coordsTable["y"][coord1.y];
75             foreach (Point coord2 in relevantCoords) {
76                 bool isRight = coord2.x > coord1.x;
77                 if (isRight) rectangleCount += clockwiseCountRectangles(
78                     coord2, coordsTable, DOWN, lowerLeftY);
79             }
80         }
81         return rectangleCount;
82     }
83 }
84
85 public class Point {
86     public int x;
87     public int y;
88
89     public Point(int x, int y) {
90         this.x = x;
91         this.y = y;
92     }
93 }
94 }
95
```

