

Solution 1Solution 2Solution 3

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 using System;
4 using System.Collections.Generic;
5
6 public class Program {
7     static string UP = "up";
8     static string RIGHT = "right";
9     static string DOWN = "down";
10    static string LEFT = "left";
11
12    // O(n^2) time | O(n^2) space - where n is the number of coordinates
13    public static int RectangleMania(Point[] coords) {
14        Dictionary<string, Dictionary<string, List<Point> > > coordsTable = getCoordsTable(
15            coords);
16        return getRectangleCount(coords, coordsTable);
17    }
18
19    public static Dictionary<string, Dictionary<string, List<Point> > > getCoordsTable(
20        Point[] coords) {
21        Dictionary<string, Dictionary<string,
22            List<Point> > > coordsTable = new Dictionary<string,
23            Dictionary<string,
24            List<Point> > >>();
25        foreach (Point coord1 in coords) {
26            Dictionary<string, List<Point> > coord1Directions = new Dictionary<string,
27                List<Point> >>();
28            coord1Directions.Add(UP, new List<Point>());
29            coord1Directions.Add(RIGHT, new List<Point>());
30            coord1Directions.Add(DOWN, new List<Point>());
31            coord1Directions.Add(LEFT, new List<Point>());
32            foreach (Point coord2 in coords) {
33                string coord2Direction = getCoordDirection(coord1, coord2);
34                if (coord1Directions.ContainsKey(coord2Direction)) coord1Directions[
35                    coord2Direction].Add(coord2);
36            }
37            string coord1string = coordToString(coord1);
38            coordsTable.Add(coord1string, coord1Directions);
39        }
40        return coordsTable;
41    }
42
43    public static string getCoordDirection(Point coord1, Point coord2) {
44        if (coord2.y == coord1.y) {
45            if (coord2.x > coord1.x) {
46                return RIGHT;
47            } else if (coord2.x < coord1.x) {
48                return LEFT;
49            }
50        } else if (coord2.x == coord1.x) {
51            if (coord2.y > coord1.y) {
52                return UP;
53            } else if (coord2.y < coord1.y) {
54                return DOWN;
55            }
56        }
57        return "";
58    }
59
60    public static int getRectangleCount(Point[] coords, Dictionary<string, Dictionary<string,
61        List<Point> > > coordsTable)
62    {
63        int rectangleCount = 0;
64        foreach (Point coord in coords) {
65            rectangleCount += clockwiseCountRectangles(coord, coordsTable, UP, coord);
66        }
67        return rectangleCount;
68    }
69
70    public static int clockwiseCountRectangles(
71        Point coord,
72        Dictionary<string, Dictionary<string, List<Point> > > coordsTable,
73        string direction,
74        Point origin
75    ) {
76        string coordstring = coordToString(coord);
77        if (direction == LEFT) {
78            bool rectangleFound = coordsTable[coordstring][LEFT].Contains(origin);
79            return rectangleFound ? 1 : 0;
80        } else {
81            int rectangleCount = 0;
82            string nextDirection = getNextClockwiseDirection(direction);
83            foreach (Point nextCoord in coordsTable[coordstring][direction]) {
84                rectangleCount += clockwiseCountRectangles(nextCoord, coordsTable,
85                    nextDirection, origin);
86            }
87            return rectangleCount;
88        }
89    }
90
91    public static string getNextClockwiseDirection(string direction) {
92        if (direction == UP) return RIGHT;
93        if (direction == RIGHT) return DOWN;
94        if (direction == DOWN) return LEFT;
95        return "";
96    }
97
98    public static string coordToString(Point coord) {
99        return coord.x.ToString() + "-" + coord.y.ToString();
100    }
101
102    public class Point {
103        public int x;
104        public int y;
105
106        public Point(int x, int y) {
107            this.x = x;
108            this.y = y;
109        }
110
111        public bool equals(object a) {
112            return this.x == ((Point) a).x && this.y == ((Point) a).y;
113        }
114    }
115 }
```

