

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1	Solution 2	Solution 3
<pre>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&lt;string, Dictionary&lt;string, List&lt;Point&gt; &gt; &gt; coordsTable = 15            new Dictionary&lt;string, Dictionary&lt;string, List&lt;Point&gt; &gt; &gt;(); 16        return getRectangleCount(coords, coordsTable); 17    } 18 19    public static Dictionary&lt;string, Dictionary&lt;string, List&lt;Point&gt; &gt; &gt; 20        buildCoordsTable(Point[] coords) { 21        Dictionary&lt;string, Dictionary&lt;string, List&lt;Point&gt; &gt; &gt; coordsTable = 22            new Dictionary&lt;string, Dictionary&lt;string, List&lt;Point&gt; &gt; &gt;(); 23        foreach (Point coord in coords) { 24            Dictionary&lt;string, List&lt;Point&gt; &gt; coordDirections = 25                new Dictionary&lt;string, List&lt;Point&gt; &gt;(); 26            coordDirections.Add(UP, new List&lt;Point&gt;()); 27            coordDirections.Add(RIGHT, new List&lt;Point&gt;()); 28            coordDirections.Add(DOWN, new List&lt;Point&gt;()); 29            coordDirections.Add(LEFT, new List&lt;Point&gt;()); 30            foreach (Point coord2 in coords) { 31                string coord2Direction = getCoordDirection(coord, coord2); 32                if (coord2Direction != null) { 33                    coordDirections[coord2Direction].Add(coord2); 34                } 35            } 36            coordsTable[coord.x + "," + coord.y] = coordDirections; 37        } 38        return coordsTable; 39    } 40 41    private static string getCoordDirection(Point coord1, Point coord2) { 42        if (coord1.x == coord2.x) { 43            if (coord1.y &lt; coord2.y) return "down"; 44            if (coord1.y &gt; coord2.y) return "up"; 45        } 46        if (coord1.y == coord2.y) { 47            if (coord1.x &lt; coord2.x) return "right"; 48            if (coord1.x &gt; coord2.x) return "left"; 49        } 50        return null; 51    } 52 53    private static int getRectangleCount(Point[] coords, Dictionary&lt;string, Dictionary&lt;string, List&lt;Point&gt; &gt; &gt; coordsTable) { 54        int count = 0; 55        foreach (string coord1 in coordsTable.Keys) { 56            foreach (string coord2 in coordsTable.Keys) { 57                if (coord1 &lt; coord2) { 58                    List&lt;Point&gt; coord1Points = coordsTable[coord1]; 59                    List&lt;Point&gt; coord2Points = coordsTable[coord2]; 60                    if (coord1Points.Count &gt; 0 &amp; coord2Points.Count &gt; 0) { 61                        count++; 62                    } 63                } 64            } 65        } 66        return count; 67    } 68 }</pre>		<pre>1 public class Program { 2     public static int RectangleMania(Point[] coords) { 3         // Write your code here. 4         return -1; 5     } 6 7     public class Point { 8         public int x; 9         public int y; 10    } 11 12    public Point(int x, int y) { 13        this.x = x; 14        this.y = y; 15    } 16 } 17</pre>

Our Tests

Custom Output

Submit Code

```
1 public class Program {
2     static int RectangleMania(Point[] coords) {
3         // Write your code here.
4         return -1;
5     }
6
7     public class Point {
8         public int x;
9         public int y;
10    }
11
12    public Point(int x, int y) {
13        this.x = x;
14        this.y = y;
15    }
16 }
```

Run or submit code when you're ready.