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

using System;
using System.Collections.Generic;

public class Program {
  // O(n^2) time | O(n) space - where n is the number of coordinates
  public static int RectangleMania(Point[] coords) {
    HashSet<string> coordsTable = getCoordsTable(coords);
    return getRectangleCount(coords, coordsTable);
  }

  public static HashSet<string> getCoordsTable(Point[] coords) {
    HashSet<string> coordsTable = new HashSet<string>();
    foreach (Point coord in coords) {
      string coordstring = coordTostring(coord);
      coordsTable.Add(coordstring);
    }
    return coordsTable;
  }

  public static int getRectangleCount(Point[] coords, HashSet<string> coordsTable) {
    int rectangleCount = 0;
    foreach (Point coord1 in coords) {
      foreach (Point coord2 in coords) {
        if (!isInUpperRight(coord1, coord2)) continue;
        string upperCoordstring = coordTostring(new Point(coord1.x,
            coord2.y));
        string rightCoordstring = coordTostring(new Point(coord2.x,
            coord1.y));
        if (
          coordsTable.Contains(upperCoordstring) &&
          coordsTable.Contains(rightCoordstring)
          ) rectangleCount++;
      }
    }
    return rectangleCount;
  }

  public static bool isInUpperRight(Point coord1, Point coord2) {
    return coord2.x > coord1.x && coord2.y > coord1.y;
  }

  public static string coordTostring(Point coord) {
    return coord.x.ToString() + "-" + coord.y.ToString();
  }

  public class Point {
    public int x;
    public int y;

    public Point(int x, int y) {
      this.x = x;
      this.y = y;
    }
  }
}
```