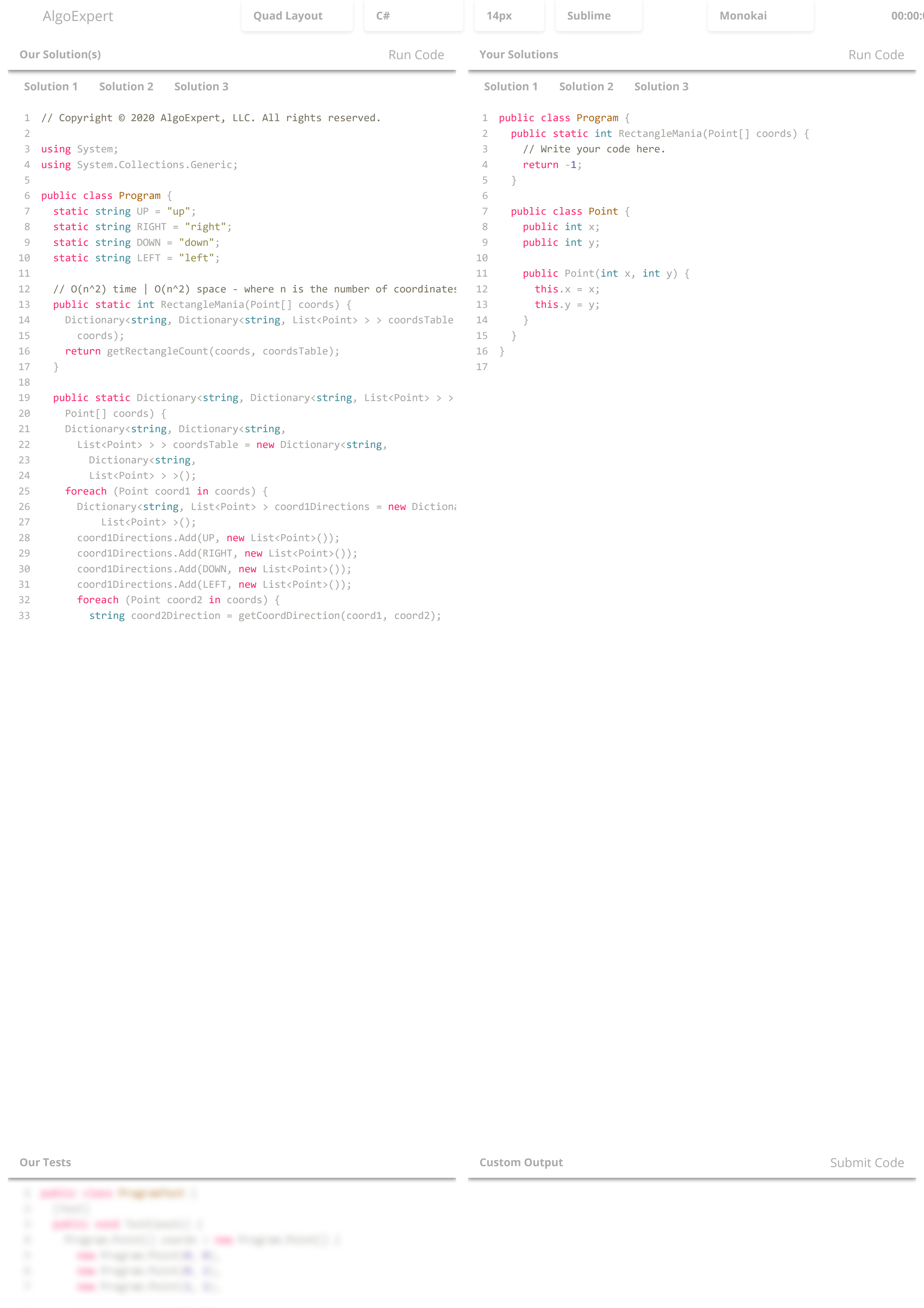

```
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

using System;
using System.Collections.Generic;

public class Program {
  static string UP = "up";
  static string RIGHT = "right";
  static string DOWN = "down";
  static string LEFT = "left";

  // O(n^2) time | O(n^2) space - where n is the number of coordinate
  public static int RectangleMania(Point[] coords) {
    Dictionary<string, Dictionary<string, List<Point> > > coordsTable
      coords);
    return getRectangleCount(coords, coordsTable);
  }

  public static Dictionary<string, Dictionary<string, List<Point> > >
    Point[] coords) {
    Dictionary<string, Dictionary<string,
      List<Point> > > coordsTable = new Dictionary<string,
        Dictionary<string,
        List<Point> > >();
    foreach (Point coord1 in coords) {
      Dictionary<string, List<Point> > coord1Directions = new Dictiona
          List<Point> >();
      coord1Directions.Add(UP, new List<Point>());
      coord1Directions.Add(RIGHT, new List<Point>());
      coord1Directions.Add(DOWN, new List<Point>());
      coord1Directions.Add(LEFT, new List<Point>());
      foreach (Point coord2 in coords) {
        string coord2Direction = getCoordDirection(coord1, coord2);
```

```
public class Program {
  public static int RectangleMania(Point[] coords) {
    // Write your code here.
    return -1;
  }

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

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

```
[illegible]
```

Run or submit code when you're ready.