AlgoExpert Quad Layout JavaScript 12px Sublime Monokai 00:00:00

Prompt Scratchpad Our Solution(s) Video Explanation Run Code

```
Solution 1
                Solution 2
                                Solution 3
 1\ \ \ //\ \mbox{Copyright @ 2020 AlgoExpert, LLC.} All rights reserved.
    // O(n^2) time | O(n^2) space - where n is the number of coordinates
    function rectangleMania(coords) {
      const coordsTable = getCoordsTable(coords);
      return getRectangleCount(coords, coordsTable);
 9
    function \ getCoordsTable(coords) \ \{
10
      const coordsTable = {};
11
      for (const coord1 of coords) {
12
         const coord1Directions = {
          [UP]: [],
13
           [RIGHT]: [],
14
15
           [DOWN]: [],
16
           [LEFT]: [],
17
         for (const coord2 of coords) {
18
19
           const coord2Direction = getCoordDirection(coord1, coord2);
20
           \textbf{if} \ (\texttt{coord2Direction} \ \textbf{in} \ \texttt{coord1Directions}) \ \texttt{coord1Directions} [\texttt{coord2Direction}]. \textbf{push} (\texttt{coord2}); \\
21
22
         const coord1String = coordToString(coord1);
23
         coordsTable[coord1String] = coord1Directions;
24
25
      return coordsTable;
26 }
27
28
     function \ getCoordDirection(coord1, \ coord2) \ \{
29
      const [x1, y1] = coord1;
30
      const [x2, y2] = coord2;
      if (y2 === y1) {
31
        if (x2 > x1) {
32
33
           return RIGHT;
34
         } else if (x2 < x1) {
35
           return LEFT;
36
       } else if (x2 === x1) {
38
        if (y2 > y1) {
39
          return UP;
         } else if (y2 < y1) {
40
41
           return DOWN;
42
43
44
      return '';
45
46
47
    function \ getRectangleCount(coords, \ coordsTable) \ \{
48
      let rectangleCount = 0;
49
      for (const coord of coords) {
50
         rectangleCount += clockwiseCountRectangles(coord, coordsTable, UP, coord);
51
52
      return rectangleCount;
53 }
54
55
     function\ clockwise Count Rectangles (coord,\ coords Table,\ direction,\ origin)\ \{
56
      const coordString = coordToString(coord);
57
      if (direction === LEFT) {
58
        const rectangleFound = coordsTable[coordString][LEFT].includes(origin);
59
        return rectangleFound ? 1 : 0;
60
      } else {
61
         let rectangleCount = 0;
62
         const nextDirection = getNextClockwiseDirection(direction);
63
         \begin{tabular}{ll} for (const \ nextCoord \ of \ coordsTable[coordString][direction]) \ \{ \end{tabular}
64
           rectangleCount += clockwiseCountRectangles(nextCoord, coordsTable, nextDirection, origin);
65
66
         return rectangleCount;
67
68 }
69
     function \ getNextClockwiseDirection(direction) \ \{
      if (direction === UP) return RIGHT;
72
      if (direction === RIGHT) return DOWN;
73
      if (direction === DOWN) return LEFT;
      return '';
74
75 }
76
77
    function coordToString(coord) {
      const [x, y] = coord;
      return `${x}-${y}`;
79
80 }
81
82 const UP = 'up';
    const RIGHT = 'right';
    const DOWN = 'down
85 const LEFT = 'left';
87 exports.rectangleMania = rectangleMania;
88
```