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

Prompt Scratchpad Our Solution(s) Video Explanation Run Code

```
Solution 3
  Solution 1 Solution 2
  1\, // Copyright @ 2020 AlgoExpert, LLC. All rights reserved.
     class Program {
          let UP = "up"
          let DOWN = "down"
          let LEFT = "left"
          let RIGHT = "right"
          func coordToString(_ coord: [Int]) -> String {
              let x = coord[0]
 10
              let y = coord[1]
 11
 12
 13
              return "(x)-(y)"
 14
 15
 16
          // O(n^2) time | O(n^2) space
 17
          func rectangleMania(_ coords: [[Int]]) -> Int {
 18
              let coordsTable = getCoordsTable(coords)
 19
              return getRectangleCount(coords, coordsTable)
 20
 21
 22
          func getCoordsTable(_ coords: [[Int]]) -> [String: [String: [[Int]]]] {
 23
              var coordsTable = [String: [String: [[Int]]]]()
 24
 25
              for coord1 in coords {
 26
                  var coord1Directions: [String: [[Int]]] = [UP: [], DOWN: [], LEFT: [], RIGHT: []]
 27
 28
                  for coord2 in coords {
 29
                       let coord2Direction = getCoordDirection(coord1, coord2)
 31
                       if var coordinatesForDirection = coord1Directions[coord2Direction] {
 32
                           coordinatesForDirection.append(coord2)
 33
                           coord1Directions[coord2Direction] = coordinatesForDirection
 34
 35
 36
 37
                  let coords1String = coordToString(coord1)
 38
                  coordsTable[coords1String] = coord1Directions
 39
 40
41
              return coordsTable
 42
 43
 44
          func getCoordDirection(_ coord1: [Int], _ coord2: [Int]) -> String {
 45
              let x1 = coord1[0]
              let y1 = coord1[1]
 46
 47
 48
              let x2 = coord2[0]
 49
              let y2 = coord2[1]
 50
 51
              if y1 == y2 {
 52
                  if x1 < x2 {
 53
                      return RIGHT
 54
                  } else {
 55
                      return LEFT
 56
 57
              } else if x1 == x2 {
 58
                  if y1 < y2 {
 59
                      return UP
 60
                  } else {
61
                       return DOWN
 62
 63
 64
              return ""
65
66
67
          \label{lem:func_getRectangleCount} \textbf{func_getRectangleCount}(\_\texttt{coords}: [[Int]]), \_\texttt{coordsTable}: [\texttt{String}: [[Int]]]) \ \rightarrow \ \texttt{Int} \ \{\texttt{String}: [[Int]]]) \ \rightarrow \ \texttt{Int} \ \{\texttt{String}: [[Int]]]\}
 68
 69
              var rectangleCount = 0
 70
 71
              for coord in coords \{
 72
                  rectangleCount += clockwiseCountRectangles(coord, coordsTable, UP, coord)
 73
 74
 75
              return rectangleCount
 76
 77
 78
          func clockwiseCountRectangles(_ coord: [Int], _ coordsTable: [String: [String: [[Int]]]], _ direction: String, _ origin: [Int]) -> Int {
 79
              let coordString = coordToString(coord)
 80
 81
              if direction == LEFT {
 82
                  if let directionsForCoordinate = coordsTable[coordString], let coordinatesForDirection = directionsForCoordinate[direction], coordinatesForDirection.contains(origin) {
 83
                       return 1
                     else {
 85
                       return 0
 86
 87
              } else {
                  var rectangleCount = 0
88
 89
                  let nextDirection = getNextClockwiseDirection(direction)
 90
 91
                  if let directionsForCoordinate = coordsTable[coordString], let coordinatesForDirection = directionsForCoordinate[direction] {
 92
                       for nextCoord in coordinatesForDirection {
                           rectangleCount += clockwiseCountRectangles(nextCoord, coordsTable, nextDirection, origin)
 93
 94
 95
 96
 97
                  return rectangleCount
 98
99
100
          func getNextClockwiseDirection(_ direction: String) -> String {
101
              if direction == UP {
102
103
                  return RIGHT
104
105
              if direction == RIGHT {
106
107
                  return DOWN
108
109
110
               \begin{tabular}{ll} \textbf{if} & direction == DOWN & \{ \end{tabular} 
111
                  return LEFT
112
113
              return ""
114
```