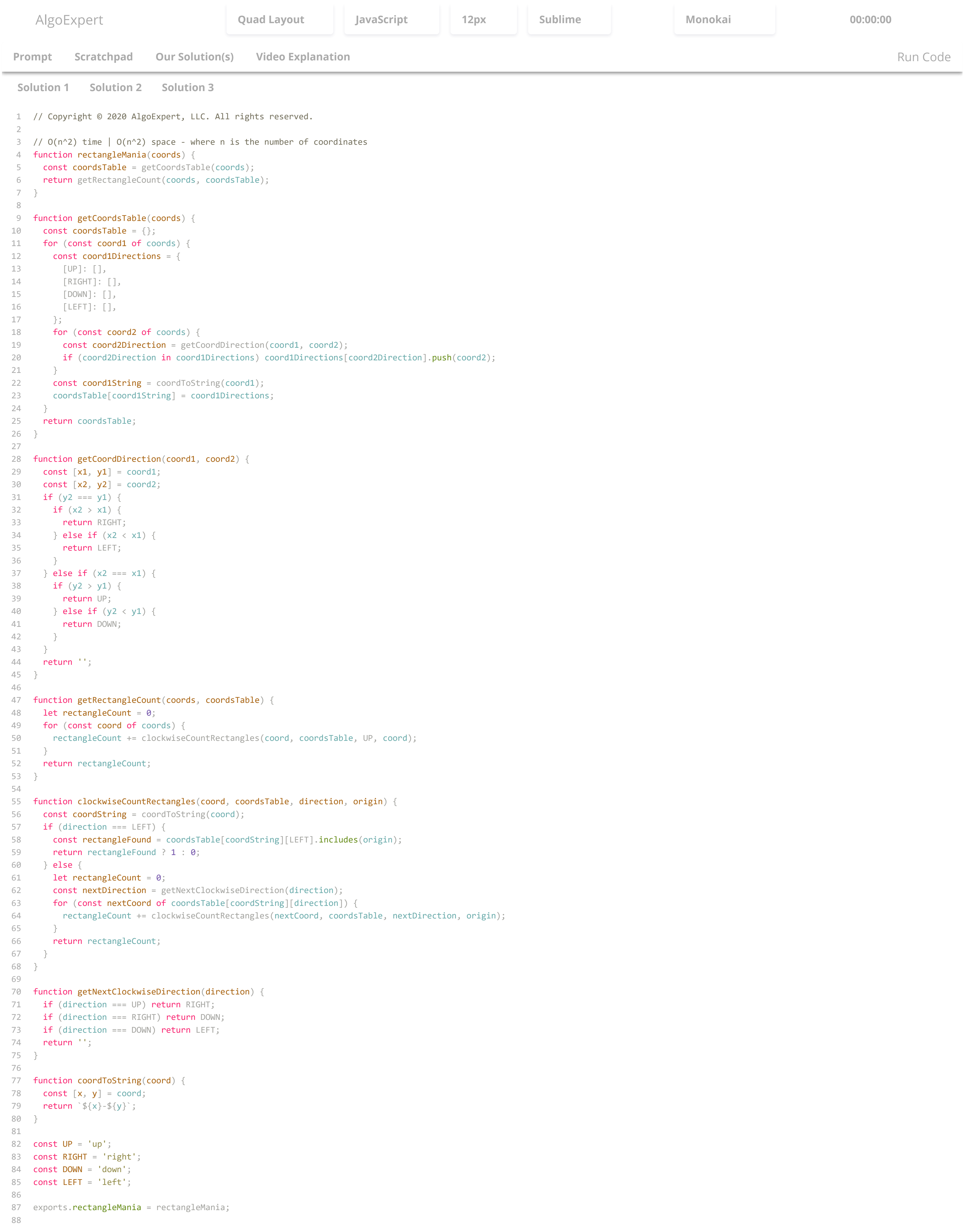

```javascript
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

// 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);
}

function getCoordsTable(coords) {
  const coordsTable = {};
  for (const coord1 of coords) {
    const coord1Directions = {
      [UP]: [],
      [RIGHT]: [],
      [DOWN]: [],
      [LEFT]: [],
    };
    for (const coord2 of coords) {
      const coord2Direction = getCoordDirection(coord1, coord2);
      if (coord2Direction in coord1Directions) coord1Directions[coord2Direction].push(coord2);
    }
    const coord1String = coordToString(coord1);
    coordsTable[coord1String] = coord1Directions;
  }
  return coordsTable;
}

function getCoordDirection(coord1, coord2) {
  const [x1, y1] = coord1;
  const [x2, y2] = coord2;
  if (y2 === y1) {
    if (x2 > x1) {
      return RIGHT;
    } else if (x2 < x1) {
      return LEFT;
    }
  } else if (x2 === x1) {
    if (y2 > y1) {
      return UP;
    } else if (y2 < y1) {
      return DOWN;
    }
  }
  return '';
}

function getRectangleCount(coords, coordsTable) {
  let rectangleCount = 0;
  for (const coord of coords) {
    rectangleCount += clockwiseCountRectangles(coord, coordsTable, UP, coord);
  }
  return rectangleCount;
}

function clockwiseCountRectangles(coord, coordsTable, direction, origin) {
  const coordString = coordToString(coord);
  if (direction === LEFT) {
    const rectangleFound = coordsTable[coordString][LEFT].includes(origin);
    return rectangleFound ? 1 : 0;
  } else {
    let rectangleCount = 0;
    const nextDirection = getNextClockwiseDirection(direction);
    for (const nextCoord of coordsTable[coordString][direction]) {
      rectangleCount += clockwiseCountRectangles(nextCoord, coordsTable, nextDirection, origin);
    }
    return rectangleCount;
  }
}

function getNextClockwiseDirection(direction) {
  if (direction === UP) return RIGHT;
  if (direction === RIGHT) return DOWN;
  if (direction === DOWN) return LEFT;
  return '';
}

function coordToString(coord) {
  const [x, y] = coord;
  return `${x}-${y}`;
}

const UP = 'up';
const RIGHT = 'right';
const DOWN = 'down';
const LEFT = 'left';

exports.rectangleMania = rectangleMania;
```