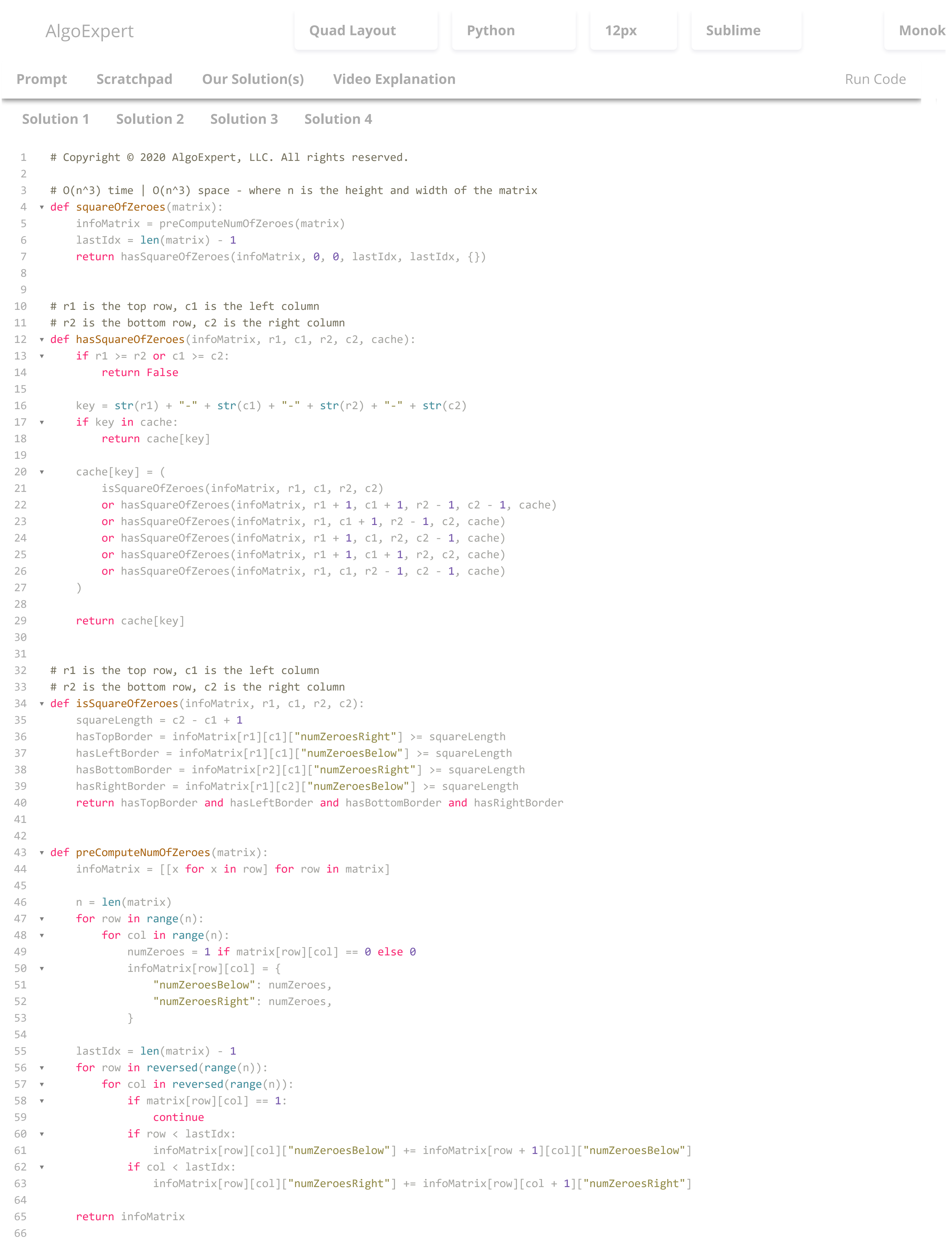

```python
# Copyright © 2020 AlgoExpert, LLC. All rights reserved.

# O(n^3) time | O(n^3) space - where n is the height and width of the matrix
def squareOfZeroes(matrix):
    infoMatrix = preComputeNumOfZeroes(matrix)
    lastIdx = len(matrix) - 1
    return hasSquareOfZeroes(infoMatrix, 0, 0, lastIdx, lastIdx, {})


# r1 is the top row, c1 is the left column
# r2 is the bottom row, c2 is the right column
def hasSquareOfZeroes(infoMatrix, r1, c1, r2, c2, cache):
    if r1 >= r2 or c1 >= c2:
        return False

    key = str(r1) + "-" + str(c1) + "-" + str(r2) + "-" + str(c2)
    if key in cache:
        return cache[key]

    cache[key] = (
        isSquareOfZeroes(infoMatrix, r1, c1, r2, c2)
        or hasSquareOfZeroes(infoMatrix, r1 + 1, c1 + 1, r2 - 1, c2 - 1, cache)
        or hasSquareOfZeroes(infoMatrix, r1, c1 + 1, r2 - 1, c2, cache)
        or hasSquareOfZeroes(infoMatrix, r1 + 1, c1, r2, c2 - 1, cache)
        or hasSquareOfZeroes(infoMatrix, r1 + 1, c1 + 1, r2, c2, cache)
        or hasSquareOfZeroes(infoMatrix, r1, c1, r2 - 1, c2 - 1, cache)
    )

    return cache[key]


# r1 is the top row, c1 is the left column
# r2 is the bottom row, c2 is the right column
def isSquareOfZeroes(infoMatrix, r1, c1, r2, c2):
    squareLength = c2 - c1 + 1
    hasTopBorder = infoMatrix[r1][c1]["numZeroesRight"] >= squareLength
    hasLeftBorder = infoMatrix[r1][c1]["numZeroesBelow"] >= squareLength
    hasBottomBorder = infoMatrix[r2][c1]["numZeroesRight"] >= squareLength
    hasRightBorder = infoMatrix[r1][c2]["numZeroesBelow"] >= squareLength
    return hasTopBorder and hasLeftBorder and hasBottomBorder and hasRightBorder


def preComputeNumOfZeroes(matrix):
    infoMatrix = [[x for x in row] for row in matrix]

    n = len(matrix)
    for row in range(n):
        for col in range(n):
            numZeroes = 1 if matrix[row][col] == 0 else 0
            infoMatrix[row][col] = {
                "numZeroesBelow": numZeroes,
                "numZeroesRight": numZeroes,
            }

    lastIdx = len(matrix) - 1
    for row in reversed(range(n)):
        for col in reversed(range(n)):
            if matrix[row][col] == 1:
                continue
            if row < lastIdx:
                infoMatrix[row][col]["numZeroesBelow"] += infoMatrix[row + 1][col]["numZeroesBelow"]
            if col < lastIdx:
                infoMatrix[row][col]["numZeroesRight"] += infoMatrix[row][col + 1]["numZeroesRight"]

    return infoMatrix

```