

Solution 1	Solution 2	Solution 3	Solution 4
<pre>1 # Copyright © 2020 AlgoExpert, LLC. All rights reserved. 2 3 # O(n^3) time O(n^2) space - where n is the height and width of the matrix 4 ▾ def squareOfZeroes(matrix): 5 infoMatrix = preComputeNumOfZeroes(matrix) 6 n = len(matrix) 7 ▾ for topRow in range(n): 8 ▾ for leftCol in range(n): 9 squareLength = 2 10 ▾ while squareLength <= n - leftCol and squareLength <= n - topRow: 11 bottomRow = topRow + squareLength - 1 12 rightCol = leftCol + squareLength - 1 13 ▾ if isSquareOfZeroes(infoMatrix, topRow, leftCol, bottomRow, rightCol): 14 return True 15 squareLength += 1 16 return False 17 18 19 # r1 is the top row, c1 is the left column 20 # r2 is the bottom row, c2 is the right column 21 ▾ def isSquareOfZeroes(infoMatrix, r1, c1, r2, c2): 22 squareLength = c2 - c1 + 1 23 hasTopBorder = infoMatrix[r1][c1]["numZeroesRight"] >= squareLength 24 hasLeftBorder = infoMatrix[r1][c1]["numZeroesBelow"] >= squareLength 25 hasBottomBorder = infoMatrix[r2][c1]["numZeroesRight"] >= squareLength 26 hasRightBorder = infoMatrix[r1][c2]["numZeroesBelow"] >= squareLength 27 return hasTopBorder and hasLeftBorder and hasBottomBorder and hasRightBorder 28 29 30 ▾ def preComputeNumOfZeroes(matrix): 31 infoMatrix = [[x for x in row] for row in matrix] 32 33 n = len(matrix) 34 ▾ for row in range(n): 35 ▾ for col in range(n): 36 numZeroes = 1 if matrix[row][col] == 0 else 0 37 ▾ infoMatrix[row][col] = { 38 "numZeroesBelow": numZeroes, 39 "numZeroesRight": numZeroes, 40 } 41 42 lastIdx = len(matrix) - 1 43 ▾ for row in reversed(range(n)): 44 ▾ for col in reversed(range(n)): 45 ▾ if matrix[row][col] == 1: 46 continue 47 ▾ if row < lastIdx: 48 infoMatrix[row][col]["numZeroesBelow"] += infoMatrix[row + 1][col]["numZeroesBelow"] 49 ▾ if col < lastIdx: 50 infoMatrix[row][col]["numZeroesRight"] += infoMatrix[row][col + 1]["numZeroesRight"] 51 52 return infoMatrix 53</pre>			

