

Prompt	Scratchpad	Our Solution(s)	Video Explanation	Run Code
--------	------------	-----------------	-------------------	----------

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^3) space - where n is the height and width of the matrix 4 ▾ def squareOfZeroes(matrix): 5 infoMatrix = preComputeNumOfZeroes(matrix) 6 lastIdx = len(matrix) - 1 7 return hasSquareOfZeroes(infoMatrix, 0, 0, lastIdx, lastIdx, {}) 8 9 10 # r1 is the top row, c1 is the left column 11 # r2 is the bottom row, c2 is the right column 12 ▾ def hasSquareOfZeroes(infoMatrix, r1, c1, r2, c2, cache): 13 ▾ if r1 >= r2 or c1 >= c2: 14 return False 15 16 key = str(r1) + "-" + str(c1) + "-" + str(r2) + "-" + str(c2) 17 ▾ if key in cache: 18 return cache[key] 19 20 ▾ cache[key] = (21 isSquareOfZeroes(infoMatrix, r1, c1, r2, c2) 22 or hasSquareOfZeroes(infoMatrix, r1 + 1, c1 + 1, r2 - 1, c2 - 1, cache) 23 or hasSquareOfZeroes(infoMatrix, r1, c1 + 1, r2 - 1, c2, cache) 24 or hasSquareOfZeroes(infoMatrix, r1 + 1, c1, r2, c2 - 1, cache) 25 or hasSquareOfZeroes(infoMatrix, r1 + 1, c1 + 1, r2, c2, cache) 26 or hasSquareOfZeroes(infoMatrix, r1, c1, r2 - 1, c2 - 1, cache) 27) 28 29 return cache[key] 30 31 32 # r1 is the top row, c1 is the left column 33 # r2 is the bottom row, c2 is the right column 34 ▾ def isSquareOfZeroes(infoMatrix, r1, c1, r2, c2): 35 squareLength = c2 - c1 + 1 36 hasTopBorder = infoMatrix[r1][c1]["numZeroesRight"] >= squareLength 37 hasLeftBorder = infoMatrix[r1][c1]["numZeroesBelow"] >= squareLength 38 hasBottomBorder = infoMatrix[r2][c1]["numZeroesRight"] >= squareLength 39 hasRightBorder = infoMatrix[r1][c2]["numZeroesBelow"] >= squareLength 40 return hasTopBorder and hasLeftBorder and hasBottomBorder and hasRightBorder 41 42 43 ▾ def preComputeNumOfZeroes(matrix): 44 infoMatrix = [[x for x in row] for row in matrix] 45 46 n = len(matrix) 47 ▾ for row in range(n): 48 ▾ for col in range(n): 49 numZeroes = 1 if matrix[row][col] == 0 else 0 50 ▾ infoMatrix[row][col] = { 51 "numZeroesBelow": numZeroes, 52 "numZeroesRight": numZeroes, 53 } 54 55 lastIdx = len(matrix) - 1 56 ▾ for row in reversed(range(n)): 57 ▾ for col in reversed(range(n)): 58 ▾ if matrix[row][col] == 1: 59 continue 60 ▾ if row < lastIdx: 61 infoMatrix[row][col]["numZeroesBelow"] += infoMatrix[row + 1][col]["numZeroesBelow"] 62 ▾ if col < lastIdx: 63 infoMatrix[row][col]["numZeroesRight"] += infoMatrix[row][col + 1]["numZeroesRight"] 64 65 return infoMatrix 66</pre>			

