

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 ▼ import java.util.*; 4 5 ▼ class Program { 6     // O(n^4) time   O(n^3) space - where n is the height and width of the matrix 7     ▼ public static boolean squareOfZeroes(List&lt;List&lt;Integer&gt;&gt; matrix) { 8         int lastIdx = matrix.size() - 1; 9         Map&lt;String, Boolean&gt; cache = new HashMap&lt;String, Boolean&gt;(); 10        return hasSquareOfZeroes(matrix, 0, 0, lastIdx, lastIdx, cache); 11    } 12 13    // r1 is the top row, c1 is the left column 14    // r2 is the bottom row, c2 is the right column 15    public static boolean hasSquareOfZeroes( 16        List&lt;List&lt;Integer&gt;&gt; matrix, 17        int r1, 18        int c1, 19        int r2, 20        int c2, 21        Map&lt;String, Boolean&gt; cache 22    ) { 23        if (r1 &gt;= r2    c1 &gt;= c2) return false; 24 25        String key = String.valueOf(r1) + '-' + String.valueOf(c1) + '-' + String.valueOf(r2) + '-' + String.valueOf(c2); 26        if (cache.containsKey(key)) return cache.get(key); 27 28        cache.put(key, 29            isSquareOfZeroes(matrix, r1, c1, r2, c2)    30            hasSquareOfZeroes(matrix, r1 + 1, c1 + 1, r2 - 1, c2 - 1, cache)    31            hasSquareOfZeroes(matrix, r1, c1 + 1, r2 - 1, c2, cache)    32            hasSquareOfZeroes(matrix, r1 + 1, c1, r2, c2 - 1, cache)    33            hasSquareOfZeroes(matrix, r1 + 1, c1 + 1, r2, c2, cache)    34            hasSquareOfZeroes(matrix, r1, c1, r2 - 1, c2 - 1, cache)); 35 36        return cache.get(key); 37    } 38 39    // r1 is the top row, c1 is the left column 40    // r2 is the bottom row, c2 is the right column 41    public static boolean isSquareOfZeroes(List&lt;List&lt;Integer&gt;&gt; matrix, 42        int r1, 43        int c1, 44        int r2, 45        int c2 46    ) { 47        ▼ for (int row = r1; row &lt; r2 + 1; row++) { 48            if (matrix.get(row).get(c1) != 0    matrix.get(row).get(c2) != 0) return false; 49        } 50        ▼ for (int col = c1; col &lt; c2 + 1; col++) { 51            if (matrix.get(r1).get(col) != 0    matrix.get(r2).get(col) != 0) return false; 52        } 53        return true; 54    } 55 } 56</pre>			

