

Solution 1Solution 2Solution 3Solution 4

```
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 ▾ function squareOfZeroes(matrix) {
5     const infoMatrix = preComputeNumOfZeroes(matrix);
6     const n = matrix.length;
7     ▾ for (let topRow = 0; topRow < n; topRow++) {
8     ▾     for (let leftCol = 0; leftCol < n; leftCol++) {
9         let squareLength = 2;
10    ▾     while (squareLength <= n - leftCol && squareLength <= n - topRow) {
11        const bottomRow = topRow + squareLength - 1;
12        const rightCol = leftCol + squareLength - 1;
13        if (isSquareOfZeroes(infoMatrix, topRow, leftCol, bottomRow, rightCol)) return true;
14        squareLength++;
15    }
16    }
17 }
18 return false;
19 }
20
21 // r1 is the top row, c1 is the left column
22 // r2 is the bottom row, c2 is the right column
23 ▾ function isSquareOfZeroes(infoMatrix, r1, c1, r2, c2) {
24     const squareLength = c2 - c1 + 1;
25     const hasTopBorder = infoMatrix[r1][c1].numZeroesRight >= squareLength;
26     const hasLeftBorder = infoMatrix[r1][c1].numZeroesBelow >= squareLength;
27     const hasBottomBorder = infoMatrix[r2][c1].numZeroesRight >= squareLength;
28     const hasRightBorder = infoMatrix[r1][c2].numZeroesBelow >= squareLength;
29     return hasTopBorder && hasLeftBorder && hasBottomBorder && hasRightBorder;
30 }
31
32 ▾ function preComputeNumOfZeroes(matrix) {
33     const infoMatrix = matrix.map(row =>
34     ▾     row.map(value => {
35         const numZeroes = value === 0 ? 1 : 0;
36         return {numZeroesBelow: numZeroes, numZeroesRight: numZeroes};
37     })),
38 );
39
40     const lastIdx = matrix.length - 1;
41     ▾ for (let row = lastIdx; row >= 0; row--) {
42     ▾     for (let col = lastIdx; col >= 0; col--) {
43         if (matrix[row][col] === 1) continue;
44     ▾     if (row < lastIdx) {
45         infoMatrix[row][col].numZeroesBelow += infoMatrix[row + 1][col].numZeroesBelow;
46     }
47     ▾     if (col < lastIdx) {
48         infoMatrix[row][col].numZeroesRight += infoMatrix[row][col + 1].numZeroesRight;
49     }
50     }
51 }
52
53 return infoMatrix;
54 }
55
56 exports.squareOfZeroes = squareOfZeroes;
57
```

