

Solution 1Solution 2Solution 3Solution 4

```
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 ▼ function squareOfZeroes(matrix) {
5     const infoMatrix = preComputeNumOfZeroes(matrix);
6     const lastIdx = matrix.length - 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 ▼ function hasSquareOfZeroes(infoMatrix, r1, c1, r2, c2, cache) {
13     if (r1 >= r2 || c1 >= c2) return false;
14
15     const key = r1.toString() + '-' + c1.toString() + '-' + r2.toString() + '-' + c2.toString();
16     if (key in cache) return cache[key];
17
18     cache[key] =
19         isSquareOfZeroes(infoMatrix, r1, c1, r2, c2) ||
20         hasSquareOfZeroes(infoMatrix, r1 + 1, c1 + 1, r2 - 1, c2 - 1, cache) ||
21         hasSquareOfZeroes(infoMatrix, r1, c1 + 1, r2 - 1, c2, cache) ||
22         hasSquareOfZeroes(infoMatrix, r1 + 1, c1, r2, c2 - 1, cache) ||
23         hasSquareOfZeroes(infoMatrix, r1 + 1, c1 + 1, r2, c2, cache) ||
24         hasSquareOfZeroes(infoMatrix, r1, c1, r2 - 1, c2 - 1, cache);
25
26     return cache[key];
27 }
28
29 // r1 is the top row, c1 is the left column
30 // r2 is the bottom row, c2 is the right column
31 ▼ function isSquareOfZeroes(infoMatrix, r1, c1, r2, c2) {
32     const squareLength = c2 - c1 + 1;
33     const hasTopBorder = infoMatrix[r1][c1].numZeroesRight >= squareLength;
34     const hasLeftBorder = infoMatrix[r1][c1].numZeroesBelow >= squareLength;
35     const hasBottomBorder = infoMatrix[r2][c1].numZeroesRight >= squareLength;
36     const hasRightBorder = infoMatrix[r1][c2].numZeroesBelow >= squareLength;
37     return hasTopBorder && hasLeftBorder && hasBottomBorder && hasRightBorder;
38 }
39
40 ▼ function preComputeNumOfZeroes(matrix) {
41     const infoMatrix = matrix.map(row =>
42     ▼     row.map(value => {
43         const numZeroes = value === 0 ? 1 : 0;
44         return {numZeroesBelow: numZeroes, numZeroesRight: numZeroes};
45     })),
46     );
47
48     const lastIdx = matrix.length - 1;
49     ▼ for (let row = lastIdx; row >= 0; row--) {
50     ▼     for (let col = lastIdx; col >= 0; col--) {
51         if (matrix[row][col] === 1) continue;
52     ▼     if (row < lastIdx) {
53         infoMatrix[row][col].numZeroesBelow += infoMatrix[row + 1][col].numZeroesBelow;
54     }
55     ▼     if (col < lastIdx) {
56         infoMatrix[row][col].numZeroesRight += infoMatrix[row][col + 1].numZeroesRight;
57     }
58     }
59     }
60
61     return infoMatrix;
62 }
63
64 exports.squareOfZeroes = squareOfZeroes;
65
```

