```
1  // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3  // O(n^4) time | O(n^3) space - where n is the height and width of the matrix
4  function squareOfZeroes(matrix) {
5    const lastIdx = matrix.length - 1;
6    return hasSquareOfZeroes(matrix, 0, 0, lastIdx, lastIdx, {});
7  }
8
9  // r1 is the top row, c1 is the left column
10 // r2 is the bottom row, c2 is the right column
11 function hasSquareOfZeroes(matrix, r1, c1, r2, c2, cache) {
12   if (r1 >= r2 || c1 >= c2) return false;
13
14   const key = r1.toString() + '-' + c1.toString() + '-' + r2.toString() + '-' + c2.toString();
15   if (key in cache) return cache[key];
16
17   cache[key] =
18     isSquareOfZeroes(matrix, r1, c1, r2, c2) ||
19     hasSquareOfZeroes(matrix, r1 + 1, c1 + 1, r2 - 1, c2 - 1, cache) ||
20     hasSquareOfZeroes(matrix, r1, c1 + 1, r2 - 1, c2, cache) ||
21     hasSquareOfZeroes(matrix, r1 + 1, c1, r2, c2 - 1, cache) ||
22     hasSquareOfZeroes(matrix, r1 + 1, c1 + 1, r2, c2, cache) ||
23     hasSquareOfZeroes(matrix, r1, c1, r2 - 1, c2 - 1, cache);
24
25   return cache[key];
26 }
27
28 // r1 is the top row, c1 is the left column
29 // r2 is the bottom row, c2 is the right column
30 function isSquareOfZeroes(matrix, r1, c1, r2, c2) {
31   for (let row = r1; row < r2 + 1; row++) {
32     if (matrix[row][c1] !== 0 || matrix[row][c2] !== 0) return false;
33   }
34   for (let col = c1; col < c2 + 1; col++) {
35     if (matrix[r1][col] !== 0 || matrix[r2][col] !== 0) return false;
36   }
37   return true;
38 }
39
40 exports.squareOfZeroes = squareOfZeroes;
41
```