

Solution 1Solution 2Solution 3Solution 4

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 using namespace std;
4
5 bool hasSquareOfZeroes(
6     vector<vector<int>>& matrix,
7     int r1,
8     int c1,
9     int r2,
10    int c2,
11    unordered_map<string, bool> &cache
12 );
13 bool isSquareOfZeroes(vector<vector<int>>& matrix, int r1, int c1, int r2, int c2);
14
15 // O(n^4) time | O(n^3) space - where n is the height and width of the matrix
16 bool squareOfZeroes(vector<vector<int>> matrix) {
17     int lastIdx = matrix.size() - 1;
18     unordered_map<string, bool> cache;
19     return hasSquareOfZeroes(matrix, 0, 0, lastIdx, lastIdx, cache);
20 }
21
22 // r1 is the top row, c1 is the left column
23 // r2 is the bottom row, c2 is the right column
24 bool hasSquareOfZeroes(
25     vector<vector<int>>& matrix,
26     int r1,
27     int c1,
28     int r2,
29     int c2,
30     unordered_map<string, bool> &cache
31 ) {
32     if (r1 >= r2 || c1 >= c2) return false;
33
34     string key = to_string(r1) + '-' + to_string(c1) + '-' + to_string(r2) + '-' + to_string(c2);
35     if (cache.find(key) != cache.end()) return cache[key];
36
37     cache[key] =
38         isSquareOfZeroes(matrix, r1, c1, r2, c2) ||
39         hasSquareOfZeroes(matrix, r1 + 1, c1 + 1, r2 - 1, c2 - 1, cache) ||
40         hasSquareOfZeroes(matrix, r1, c1 + 1, r2 - 1, c2, cache) ||
41         hasSquareOfZeroes(matrix, r1 + 1, c1, r2, c2 - 1, cache) ||
42         hasSquareOfZeroes(matrix, r1 + 1, c1 + 1, r2, c2, cache) ||
43         hasSquareOfZeroes(matrix, r1, c1, r2 - 1, c2 - 1, cache);
44
45     return cache[key];
46 }
47
48 // r1 is the top row, c1 is the left column
49 // r2 is the bottom row, c2 is the right column
50 bool isSquareOfZeroes(vector<vector<int>>& matrix, int r1, int c1, int r2, int c2) {
51     for (int row = r1; row < r2 + 1; row++) {
52         if (matrix[row][c1] != 0 || matrix[row][c2] != 0) return false;
53     }
54     for (int col = c1; col < c2 + 1; col++) {
55         if (matrix[r1][col] != 0 || matrix[r2][col] != 0) return false;
56     }
57     return true;
58 }
```

