Solution 1   Solution 2   Solution 3   Solution 4

```swift
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

class Program {
  // O(n^4) time | O(n^3) space - where n is the height and width of the matrix
  static func squareOfZeroes(_ matrix: [[Int]]) -> Bool {
    var lastIdx = matrix.count - 1
    var mat = matrix
    var cache = [String: Bool]()
    return hasSquareOfZeroes(&mat, 0, 0, lastIdx, lastIdx, &cache)
  }

  // r1 is the top row, c1 is the left column
  // r2 is the bottom row, c2 is the right column
  static func hasSquareOfZeroes(_ matrix: inout [[Int]], _ r1: Int, _ c1: Int,
                                _ r2: Int, _ c2: Int, _ cache: inout [String: Bool]) -> Bool {
    if r1 >= r2 || c1 >= c2 {
      return false
    }

    let key = String(r1) + "-" + String(c1) + "-" + String(r2) + "-" + String(c2)
    if let out = cache[key] {
      return out
    }

    let out = isSquareOfZeroes(&matrix, r1, c1, r2, c2) ||
      hasSquareOfZeroes(&matrix, r1 + 1, c1 + 1, r2 - 1, c2 - 1, &cache) ||
      hasSquareOfZeroes(&matrix, r1, c1 + 1, r2 - 1, c2, &cache) ||
      hasSquareOfZeroes(&matrix, r1 + 1, c1, r2, c2 - 1, &cache) ||
      hasSquareOfZeroes(&matrix, r1 + 1, c1 + 1, r2, c2, &cache) ||
      hasSquareOfZeroes(&matrix, r1, c1, r2 - 1, c2 - 1, &cache)
    cache[key] = out
    return out
  }

  // r1 is the top row, c1 is the left column
  // r2 is the bottom row, c2 is the right column
  static func isSquareOfZeroes(_ matrix: inout [[Int]], _ r1: Int, _ c1: Int, _ r2: Int, _ c2: Int) -> Bool {
    for row in r1 ..< r2 + 1 {
      if matrix[row][c1] != 0 || matrix[row][c2] != 0 {
        return false
      }
    }

    for col in c1 ..< c2 + 1 {
      if matrix[r1][col] != 0 || matrix[r2][col] != 0 {
        return false
      }
    }
    return true
  }
}
```