AlgoExpert Quad Layout Java 12px Sublime Monok

Prompt Scratchpad Our Solution(s) Video Explanation

Solution 2

Solution 1

Solution 3

Solution 4

```
Run Code
```

```
1
     // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
 2
 3
    ▼ import java.util.*;
 4
 5
   ▼ class Program {
       // O(n^3) time \mid O(n^3) space - where n is the height and width of the matrix
 6
       public static boolean squareOfZeroes(List<List<Integer>> matrix) {
 7
         List<List<InfoMatrixItem>> infoMatrix = preComputedNumOfZeroes(matrix);
 9
         int lastIdx = matrix.size() - 1;
         Map<String, Boolean> cache = new HashMap<String, Boolean>();
10
11
         return hasSquareOfZeroes(infoMatrix, 0, 0, lastIdx, lastIdx, cache);
12
13
14
       // r1 is the top row, c1 is the left column
15
       // r2 is the bottom row, c2 is the right column
16
       public static boolean hasSquareOfZeroes(
17
         List<List<InfoMatrixItem>> matrix,
18
         int r1,
19
         int c1,
20
         int r2,
21
         int c2,
         Map<String, Boolean> cache
22
23 ▼
         if (r1 >= r2 \mid | c1 >= c2) return false;
24
25
         String key = String.valueOf(r1) + '-' + String.valueOf(c1) + '-' + String.valueOf(r2) + '-' + String.valueOf(c2);
26
          if (cache.containsKey(key)) return cache.get(key);
27
28
29
         cache.put(key,
30
            isSquareOfZeroes(matrix, r1, c1, r2, c2) ||
31
           hasSquareOfZeroes(matrix, r1 + 1, c1 + 1, r2 - 1, c2 - 1, cache) ||
            hasSquareOfZeroes(matrix, r1, c1 + 1, r2 - 1, c2, cache)
32
33
            hasSquareOfZeroes(matrix, r1 + 1, c1, r2, c2 - 1, cache) ||
            hasSquareOfZeroes(matrix, r1 + 1, c1 + 1, r2, c2, cache) ||
34
35
            hasSquareOfZeroes(matrix, r1, c1, r2 - 1, c2 - 1, cache));
36
37
         return cache.get(key);
38
       }
39
       // r1 is the top row, c1 is the left column
40
41
       // r2 is the bottom row, c2 is the right column
       public static boolean isSquareOfZeroes(List<List<InfoMatrixItem>> infoMatrix,
42
43
         int r1,
44
         int c1,
45
         int r2,
46
         int c2
47 ▼
       ) {
48
          int squareLength = c2 - c1 + 1;
49
         boolean hasTopBorder = infoMatrix.get(r1).get(c1).numZeroesRight >= squareLength;
50
         boolean hasLeftBorder = infoMatrix.get(r1).get(c1).numZeroesBelow >= squareLength;
51
         boolean hasBottomBorder = infoMatrix.get(r2).get(c1).numZeroesRight >= squareLength;
         boolean hasRightBorder = infoMatrix.get(r1).get(c2).numZeroesBelow >= squareLength;
52
         return hasTopBorder && hasLeftBorder && hasBottomBorder && hasRightBorder;
53
54
55
56 ▼
       public static List<List<InfoMatrixItem>> preComputedNumOfZeroes(List<List<Integer>> matrix) {
57
         List<List<InfoMatrixItem>> infoMatrix = new ArrayList<List<InfoMatrixItem>>();
58 ▼
          for (int i = 0; i < matrix.size(); i++) {</pre>
            List<InfoMatrixItem> inner = new ArrayList<InfoMatrixItem>();
59
            for (int j = 0; j < matrix.get(i).size(); j++) {</pre>
61
              int numZeroes = matrix.get(i).get(j) == 0 ? 1 : 0;
              inner.add(new InfoMatrixItem(numZeroes, numZeroes));
62
63
            infoMatrix.add(inner);
64
65
         }
66
67
          int lastIdx = matrix.size() - 1;
          for (int row = lastIdx; row >= 0; row--) {
68
           for (int col = lastIdx; col >= 0; col--) {
69
             if (matrix.get(row).get(col) == 1) continue;
70
71 ▼
             if (row < lastIdx) {</pre>
72
               infoMatrix.get(row).get(col).numZeroesBelow += infoMatrix.get(row + 1).get(col).numZeroesBelow;
73
74 ▼
             if (col < lastIdx) {</pre>
                infoMatrix.get(row).get(col).numZeroesRight += infoMatrix.get(row).get(col + 1).numZeroesRight;
75
76
77
78
79
          return infoMatrix;
81
82
       static class InfoMatrixItem {
```

```
public int numZeroesBelow;
public int numZeroesRight;

public int numZeroesRight;

public InfoMatrixItem(int numZeroesBelow, int numZeroesRight) {
    this.numZeroesBelow = numZeroesBelow;
    this.numZeroesRight = numZeroesRight;
}

you    }

you    }

you    }

you    int numZeroesRight) {
    this.numZeroesRight = numZeroesRight;
    }

you    }

you    int numZeroesRight) {
    this.numZeroesRight = numZeroesRight;
    }
```