1.7 Write an algorithm such that if an element in an MxN matrix is 0, its entire row and column is set to 0.

pg 48

SOLUTION

At first glance, this problem seems easy: just iterate through the matrix and every time we see a 0, set that row and column to 0. There's one problem with that solution though: we will "recognize" those 0s later on in our iteration and then set their row and column to zero. Pretty soon, our entire matrix is 0s!

One way around this is to keep a second matrix which flags the 0 locations. We would then do a second pass through the matrix to set the zeros. This would take O(MN) space.

Do we really need O(MN) space? No. Since we're going to set the entire row and column to zero, do we really need to track *which* cell in a row is zero? No. We only need to know that row 2, for example, has a zero.

The code below implement this algorithm. We keep track in two arrays all the rows with zeros and all the columns with zeros. We then make a second pass of the matrix and set a cell to zero if its row or column is zero.

```
1   public static void setZeros(int[][] matrix) {
2       int[] row = new int[matrix.length];
3       int[] column = new int[matrix[0].length];
4       // Store the row and column index with value 0
5       for (int i = 0; i < matrix.length; i++) {
6           for (int j = 0; j < matrix[0].length;j++) {
7               if (matrix[i][j] == 0) {
8                   row[i] = 1;
9                   column[j] = 1;
10              }
11          }
12      }
13
14      // Set arr[i][j] to 0 if either row i or column j has a 0
15      for (int i = 0; i < matrix.length; i++) {
16          for (int j = 0; j < matrix[0].length; j++) {
17              if ((row[i] == 1 || column[j] == 1)) {
18                  matrix[i][j] = 0;
19              }
20          }
21      }
22  }
```