2.Medium\11.Set_matrix_0's.cpp

```
1   /*
2   QUESTION:
3   Given an m x n integer matrix matrix, if an element is 0, set its entire row and column to
    0's.
4
5   Example 1:
6   Input: matrix = [[1,1,1],[1,0,1],[1,1,1]]
7   Output: [[1,0,1],[0,0,0],[1,0,1]]
8
9   Example 2:
10  Input: matrix = [[0,1,2,0],[3,4,5,2],[1,3,1,5]]
11  Output: [[0,0,0,0],[0,4,5,0],[0,3,1,0]]
12
13  APPROACH:
14  To solve this problem in-place, we can follow these steps:
15  1. Use two boolean variables, firstRowZero and firstColZero, to check if the first row and
    first column contain zeros initially.
16  2. Iterate through the matrix and if an element is zero, set the corresponding element in the
    first row and first column to zero.
17  3. Iterate through the matrix again, excluding the first row and first column. If an element
    in the first row or first column is zero, set the current element to zero.
18  4. Finally, based on the values in firstRowZero and firstColZero, set the first row and first
    column to zero if needed.
19
20  TIME COMPLEXITY: O(m * n), where m and n are the dimensions of the matrix.
21  SPACE COMPLEXITY: O(1), as we are using constant extra space.
22
23  */
24
25  // CODE:
26  void setZeroes(vector<vector<int>>& matrix) {
27      int m = matrix.size();
28      int n = matrix[0].size();
29      bool firstRowZero = false;
30      bool firstColZero = false;
31
32      // Check if the first row contains zero
33      for (int j = 0; j < n; j++) {
34          if (matrix[0][j] == 0) {
35              firstRowZero = true;
36              break;
37          }
38      }
39
40      // Check if the first column contains zero
41      for (int i = 0; i < m; i++) {
42          if (matrix[i][0] == 0) {
43              firstColZero = true;
44              break;
45          }
46      }
47
```

```cpp
        // Mark zeros in the first row and column
        for (int i = 1; i < m; i++) {
            for (int j = 1; j < n; j++) {
                if (matrix[i][j] == 0) {
                    matrix[i][0] = 0;
                    matrix[0][j] = 0;
                }
            }
        }

        // Set rows to zero
        for (int i = 1; i < m; i++) {
            if (matrix[i][0] == 0) {
                for (int j = 1; j < n; j++) {
                    matrix[i][j] = 0;
                }
            }
        }

        // Set columns to zero
        for (int j = 1; j < n; j++) {
            if (matrix[0][j] == 0) {
                for (int i = 1; i < m; i++) {
                    matrix[i][j] = 0;
                }
            }
        }

        // Set first row to zero
        if (firstRowZero) {
            for (int j = 0; j < n; j++) {
                matrix[0][j] = 0;
            }
        }

        // Set first column to zero
        if (firstColZero) {
            for (int i = 0; i < m; i++) {
                matrix[i][0] = 0;
            }
        }
}

// TIME COMPLEXITY: O(m * n), where m and n are the dimensions of the matrix.
// SPACE COMPLEXITY: O(1), as we are using constant extra space.
```