Search2DMatrix.java

```
1   package BinarySearch;
2
3   import java.util.ArrayList;
4
5   public class Search2DMatrix {
6
7           public static boolean searchMatrix(ArrayList<ArrayList<Integer>> matrix, int target) {
8           int n = matrix.size(), m = matrix.get(0).size();
9
10          // traverse the matrix:
11 2        for (int i = 0; i < n; i++) {
12 2            for (int j = 0; j < m; j++) {
13 1                if (matrix.get(i).get(j) == target)
14 1                    return true;
15              }
16          }
17 1        return false;
18      }
19           public static boolean binarySearch(ArrayList<Integer> nums, int target) {
20                  int n = nums.size(); //size of the array
21 1                int low = 0, high = n - 1;
22
23                  // Perform the steps:
24 2                while (low <= high) {
25 2                    int mid = (low + high) / 2;
26 2                    if (nums.get(mid) == target) return true;
27 3                    else if (target > nums.get(mid)) low = mid + 1;
28 1                    else high = mid - 1;
29                  }
30 1                return false;
31              }
32
33          public static boolean searchMatrix1(ArrayList<ArrayList<Integer>> matrix, int target) {
34                  int n = matrix.size();
35                  int m = matrix.get(0).size();
36
37 2                for (int i = 0; i < n; i++) {
38 5                    if (matrix.get(i).get(0) <= target && target <= matrix.get(i).get(m - 1)) {
39 2                        return binarySearch(matrix.get(i), target);
40                      }
41                  }
42 1                return false;
43              }
44
45          public static boolean searchMatrix2(ArrayList<ArrayList<Integer>> matrix, int target) {
46                  int n = matrix.size();
47                  int m = matrix.get(0).size();
48
49                  //apply binary search:
50 2                int low = 0, high = n * m - 1;
51 2                while (low <= high) {
52 2                    int mid = (low + high) / 2;
53 2                    int row = mid / m, col = mid % m;
54 2                    if (matrix.get(row).get(col) == target) return true;
55 3                    else if (matrix.get(row).get(col) < target) low = mid + 1;
56 1                    else high = mid - 1;
57                  }
58 1                return false;
59              }
60  }
```

Mutations

```
11  1. negated conditional → KILLED
    2. changed conditional boundary → KILLED
12  1. changed conditional boundary → KILLED
    2. negated conditional → KILLED
13  1. negated conditional → KILLED
```

14	1. replaced boolean return with false for BinarySearch/Search2DMatrix::searchMatrix → KILLED
17	1. replaced boolean return with true for BinarySearch/Search2DMatrix::searchMatrix → KILLED
21	1. Replaced integer subtraction with addition → KILLED
24	1. negated conditional → KILLED 2. changed conditional boundary → KILLED
25	1. Replaced integer addition with subtraction → KILLED 2. Replaced integer division with multiplication → KILLED
26	1. negated conditional → KILLED 2. replaced boolean return with false for BinarySearch/Search2DMatrix::binarySearch → KILLED
27	1. changed conditional boundary → SURVIVED 2. negated conditional → KILLED 3. Replaced integer addition with subtraction → TIMED_OUT
28	1. Replaced integer subtraction with addition → NO_COVERAGE
30	1. replaced boolean return with true for BinarySearch/Search2DMatrix::binarySearch → NO_COVERAGE
37	1. changed conditional boundary → SURVIVED 2. negated conditional → KILLED
38	1. Replaced integer subtraction with addition → KILLED 2. negated conditional → KILLED 3. changed conditional boundary → KILLED 4. negated conditional → KILLED 5. changed conditional boundary → KILLED
39	1. replaced boolean return with false for BinarySearch/Search2DMatrix::searchMatrix1 → KILLED 2. replaced boolean return with true for BinarySearch/Search2DMatrix::searchMatrix1 → SURVIVED
42	1. replaced boolean return with true for BinarySearch/Search2DMatrix::searchMatrix1 → NO_COVERAGE
50	1. Replaced integer subtraction with addition → KILLED 2. Replaced integer multiplication with division → KILLED
51	1. negated conditional → KILLED 2. changed conditional boundary → KILLED
52	1. Replaced integer addition with subtraction → KILLED 2. Replaced integer division with multiplication → KILLED
53	1. Replaced integer division with multiplication → KILLED 2. Replaced integer modulus with multiplication → KILLED
54	1. negated conditional → KILLED 2. replaced boolean return with false for BinarySearch/Search2DMatrix::searchMatrix2 → KILLED
55	1. negated conditional → KILLED 2. changed conditional boundary → SURVIVED 3. Replaced integer addition with subtraction → TIMED_OUT
56	1. Replaced integer subtraction with addition → SURVIVED
58	1. replaced boolean return with true for BinarySearch/Search2DMatrix::searchMatrix2 → KILLED

Active mutators

- CONDITIONALS_BOUNDARY
- EMPTY_RETURNS
- FALSE_RETURNS
- INCREMENTS
- INVERT_NEGS
- MATH
- NEGATE_CONDITIONALS
- NULL_RETURNS
- PRIMITIVE_RETURNS
- TRUE_RETURNS
- VOID_METHOD_CALLS

Tests examined

- BinarySearch.Search2DMatrixTest.test(BinarySearch.Search2DMatrixTest) (0 ms)

Report generated by PIT 1.15.0