BinarySearch2dArray.java

```
1  package com.example;
2
3  public class BinarySearch2dArray {
4
5      static int[] BinarySearch(int[][] arr, int target) {
6          int rowCount = arr.length, colCount = arr[0].length;
7
8  1       if (rowCount == 1) {
9  1           return binarySearch(arr, target, 0, 0, colCount);
10         }
11
12 2       int startRow = 0, endRow = rowCount - 1, midCol = colCount / 2;
13
14 3       while (startRow < endRow - 1) {
15 3           int midRow = startRow + (endRow - startRow) / 2; // getting the index of middle row
16
17 1           if (arr[midRow][midCol] == target) {
18 1               return new int[] {midRow, midCol};
19 2           } else if (arr[midRow][midCol] < target)
20                 startRow = midRow;
21             else
22                 endRow = midRow;
23         }
24         /*
25             if the above search fails to find the target element, these conditions will be used to
26            find the target element, which further uses the binary search algorithm in the places
27            which were left unexplored.
28              */
29 1       if (arr[startRow][midCol] == target)
30 1           return new int[] {
31                 startRow,
32                 midCol,
33             };
34
35 2       if (arr[endRow][midCol] == target) return new int[] {endRow, midCol};
36
37 5       if (target <= arr[startRow][midCol - 1]) return binarySearch(arr, target, startRow, 0, midCol - 1);
38
39 9       if (target >= arr[startRow][midCol + 1] && target <= arr[startRow][colCount - 1]) return binarySearch(arr, target, startRow, midCol + 1, colCount - 1);
40
41 3       if (target <= arr[endRow][midCol - 1])
42 2           return binarySearch(arr, target, endRow, 0, midCol - 1);
43         else
44 3           return binarySearch(arr, target, endRow, midCol + 1, colCount - 1);
45     }
46
47     static int[] binarySearch(int[][] arr, int target, int row, int colStart, int colEnd) {
48 2       while (colStart <= colEnd) {
49 3           int midIndex = colStart + (colEnd - colStart) / 2;
50
51 1           if (arr[row][midIndex] == target)
52 1               return new int[] {
53                     row,
54                     midIndex,
55                 };
56 2           else if (arr[row][midIndex] < target)
57 1               colStart = midIndex + 1;
58             else
59 1               colEnd = midIndex - 1;
60         }
61
62 1       return new int[] {-1, -1};
63     }
64 }
```

Mutations

```
8   1. negated conditional → KILLED
9   1. replaced return value with null for com/example/BinarySearch2dArray::BinarySearch → KILLED
12  1. Replaced integer subtraction with addition → KILLED
    2. Replaced integer division with multiplication → KILLED
    1. changed conditional boundary → TIMED_OUT
14  2. Replaced integer subtraction with addition → TIMED_OUT
    3. negated conditional → KILLED
    1. Replaced integer subtraction with addition → SURVIVED
15  2. Replaced integer division with multiplication → KILLED
    3. Replaced integer addition with subtraction → KILLED
17  1. negated conditional → KILLED
18  1. replaced return value with null for com/example/BinarySearch2dArray::BinarySearch → KILLED
19  1. changed conditional boundary → SURVIVED
    2. negated conditional → KILLED
29  1. negated conditional → KILLED
30  1. replaced return value with null for com/example/BinarySearch2dArray::BinarySearch → NO_COVERAGE
35  1. negated conditional → KILLED
    2. replaced return value with null for com/example/BinarySearch2dArray::BinarySearch → KILLED
    1. changed conditional boundary → KILLED
    2. Replaced integer subtraction with addition → KILLED
37  3. Replaced integer subtraction with addition → SURVIVED
    4. negated conditional → KILLED
    5. replaced return value with null for com/example/BinarySearch2dArray::BinarySearch → KILLED
    1. changed conditional boundary → KILLED
    2. changed conditional boundary → KILLED
    3. Replaced integer addition with subtraction → SURVIVED
    4. Replaced integer subtraction with addition → KILLED
39  5. Replaced integer addition with subtraction → SURVIVED
    6. Replaced integer subtraction with addition → KILLED
    7. negated conditional → KILLED
    8. negated conditional → KILLED
    9. replaced return value with null for com/example/BinarySearch2dArray::BinarySearch → KILLED
    1. changed conditional boundary → KILLED
41  2. Replaced integer subtraction with addition → SURVIVED
    3. negated conditional → KILLED
    1. Replaced integer subtraction with addition → SURVIVED
42  2. replaced return value with null for com/example/BinarySearch2dArray::BinarySearch → KILLED
    1. Replaced integer addition with subtraction → SURVIVED
44  2. Replaced integer subtraction with addition → KILLED
    3. replaced return value with null for com/example/BinarySearch2dArray::BinarySearch → KILLED
48  1. changed conditional boundary → KILLED
    2. negated conditional → KILLED
    1. Replaced integer subtraction with addition → TIMED_OUT
49  2. Replaced integer division with multiplication → TIMED_OUT
    3. Replaced integer addition with subtraction → KILLED
51  1. negated conditional → KILLED
52  1. replaced return value with null for com/example/BinarySearch2dArray::binarySearch → KILLED
56  1. changed conditional boundary → SURVIVED
    2. negated conditional → KILLED
57  1. Replaced integer addition with subtraction → TIMED_OUT
59  1. Replaced integer subtraction with addition → SURVIVED
62  1. replaced return value with null for com/example/BinarySearch2dArray::binarySearch → KILLED
```

Active mutators

- BOOLEAN_FALSE_RETURN
- BOOLEAN_TRUE_RETURN
- CONDITIONALS_BOUNDARY_MUTATOR
- EMPTY_RETURN_VALUES
- INCREMENTS_MUTATOR
- INVERT_NEGS_MUTATOR
- MATH_MUTATOR
- NEGATE_CONDITIONALS_MUTATOR
- NULL_RETURN_VALUES
- PRIMITIVE_RETURN_VALS_MUTATOR
- VOID_METHOD_CALL_MUTATOR

Tests examined

- com.example.BinarySearch2dArrayTest.testBinarySearch2dArrayMiddleSide(com.example.BinarySearch2dArrayTest) (1 ms)
- com.example.BinarySearch2dArrayTest.testBinarySearch2dArrayTargetInMiddle(com.example.BinarySearch2dArrayTest) (0 ms)
- com.example.BinarySearch2dArrayTest.testBinarySearch2dArrayOneRow(com.example.BinarySearch2dArrayTest) (1 ms)
- com.example.BinarySearch2dArrayTest.testBinarySearch2dArrayUpper(com.example.BinarySearch2dArrayTest) (0 ms)
- com.example.BinarySearch2dArrayTest.testBinarySearch2dArrayNotFound(com.example.BinarySearch2dArrayTest) (0 ms)
- com.example.BinarySearch2dArrayTest.testBinarySearch2dArrayUpperSide(com.example.BinarySearch2dArrayTest) (0 ms)
- com.example.BinarySearch2dArrayTest.testBinarySearch2dArrayLowerSide(com.example.BinarySearch2dArrayTest) (1 ms)
- com.example.BinarySearch2dArrayTest.testBinarySearch2dArrayLower(com.example.BinarySearch2dArrayTest) (0 ms)
- com.example.BinarySearch2dArrayTest.testBinarySearch2dArrayMiddle(com.example.BinarySearch2dArrayTest) (0 ms)