BinarySearch2dArray.java

```
   package DivideAndConquerTesting;

   public class BinarySearch2dArray {

           static int[] BinarySearch(int[][] arr, int target) {
                   int rowCount = arr.length, colCount = arr[0].length;

1                  if (rowCount == 1) {
1                          return binarySearch(arr, target, 0, 0, colCount);
                   }

2                  int startRow = 0, endRow = rowCount - 1, midCol = colCount / 2;

3                  while (startRow < endRow - 1) {
3                          int midRow = startRow + (endRow - startRow) / 2; //getting the index of middle row

1                          if (arr[midRow][midCol] == target) {
1                                  return new int[] { midRow, midCol };
2                          } else if (arr[midRow][midCol] < target)
                                   startRow = midRow;
                           else endRow = midRow;
                   }
1                  if (arr[startRow][midCol] == target) return new int[] {
                                   startRow,
1                                  midCol,
                   };

2                  if (arr[endRow][midCol] == target) return new int[] { endRow, midCol };

4                  if (target <= arr[startRow][midCol - 1]) return binarySearch(arr,
                                   target,
                                   startRow,
                                   0,
1                                  midCol - 1
                                   );

                   if (
3                                  target >= arr[startRow][midCol + 1] &&
3                                  target <= arr[startRow][colCount - 1]
3                                  ) return binarySearch(arr, target, startRow, midCol + 1, colCount - 1);

4                  if (target <= arr[endRow][midCol - 1]) return binarySearch(
                                   arr,
                                   target,
                                   endRow,
                                   0,
1                                  midCol - 1
1                                  ); else return binarySearch(
                                                   arr,
                                                   target,
                                                   endRow,
1                                                  midCol + 1,
1                                                  colCount - 1
                                                   );
           }

           static int[] binarySearch(
                           int[][] arr,
                           int target,
                           int row,
                           int colStart,
                           int colEnd
                           ) {
2                  while (colStart <= colEnd) {
```

```
65 3                            int midIndex = colStart + (colEnd - colStart) / 2;
66  
67 1                            if (arr[row][midIndex] == target) return new int[] {
68                                              row,
69 1                                            midIndex,
70 2                            }; else if (arr[row][midIndex] < target) colStart =
71 2                            midIndex + 1; else colEnd = midIndex - 1;
72                      }
73  
74 1                    return new int[] { -1, -1 };
75              }
76      }
```

Mutations

```
8   1. negated conditional → KILLED
9   1. replaced return value with null for DivideAndConquerTesting/BinarySearch2dArray::BinarySearch → KILLED
12  1. Replaced integer subtraction with addition → KILLED
    2. Replaced integer division with multiplication → KILLED
14  1. changed conditional boundary → TIMED_OUT
    2. Replaced integer subtraction with addition → TIMED_OUT
    3. negated conditional → KILLED
15  1. Replaced integer subtraction with addition → KILLED
    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 DivideAndConquerTesting/BinarySearch2dArray::BinarySearch → KILLED
19  1. changed conditional boundary → SURVIVED
    2. negated conditional → KILLED
23  1. negated conditional → KILLED
25  1. replaced return value with null for DivideAndConquerTesting/BinarySearch2dArray::BinarySearch →
    NO_COVERAGE
28  1. negated conditional → KILLED
    2. replaced return value with null for DivideAndConquerTesting/BinarySearch2dArray::BinarySearch →
    NO_COVERAGE
30  1. changed conditional boundary → KILLED
    2. Replaced integer subtraction with addition → SURVIVED
    3. negated conditional → KILLED
    4. replaced return value with null for DivideAndConquerTesting/BinarySearch2dArray::BinarySearch → KILLED
34  1. Replaced integer subtraction with addition → SURVIVED
38  1. changed conditional boundary → SURVIVED
    2. Replaced integer addition with subtraction → SURVIVED
    3. negated conditional → KILLED
39  1. changed conditional boundary → KILLED
    2. Replaced integer subtraction with addition → KILLED
    3. negated conditional → KILLED
40  1. Replaced integer addition with subtraction → SURVIVED
    2. Replaced integer subtraction with addition → SURVIVED
    3. replaced return value with null for DivideAndConquerTesting/BinarySearch2dArray::BinarySearch → KILLED
42  1. changed conditional boundary → SURVIVED
    2. Replaced integer subtraction with addition → SURVIVED
    3. negated conditional → KILLED
    4. replaced return value with null for DivideAndConquerTesting/BinarySearch2dArray::BinarySearch → KILLED
47  1. Replaced integer subtraction with addition → SURVIVED
48  1. replaced return value with null for DivideAndConquerTesting/BinarySearch2dArray::BinarySearch → KILLED
52  1. Replaced integer addition with subtraction → SURVIVED
53  1. Replaced integer subtraction with addition → KILLED
64  1. changed conditional boundary → KILLED
    2. negated conditional → KILLED
65  1. Replaced integer subtraction with addition → TIMED_OUT
    2. Replaced integer division with multiplication → TIMED_OUT
    3. Replaced integer addition with subtraction → KILLED
67  1. negated conditional → KILLED
69  1. replaced return value with null for DivideAndConquerTesting/BinarySearch2dArray::binarySearch → KILLED
70  1. changed conditional boundary → SURVIVED
    2. negated conditional → KILLED
71  1. Replaced integer addition with subtraction → TIMED_OUT
    2. Replaced integer subtraction with addition → TIMED_OUT
74  1. replaced return value with null for DivideAndConquerTesting/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

- DivideAndConquerTesting.AllDivideConquerTesting.[engine:junit-jupiter]/[class:DivideAndConquerTesting.AllDivideConquerTesting]/[method:BinarySearch2dArrayTestMiddle()] (28 ms)

Report generated by PIT 1.6.8