

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 import java.util.*;
4
5 // O(n) time | O(n) space - where n is the total number of elements in the two-dimensional array
6 class Program {
7     public static List<Integer> zigzagTraverse(List<List<Integer>>> array) {
8         int height = array.size() - 1;
9         int width = array.get(0).size() - 1;
10        List<Integer> result = new ArrayList<Integer>();
11        int row = 0;
12        int col = 0;
13        boolean goingDown = true;
14        while (!isOutOfBounds(row, col, height, width)) {
15            result.add(array.get(row).get(col));
16            if (goingDown) {
17                if (col == 0 || row == height) {
18                    goingDown = false;
19                    if (row == height) {
20                        col++;
21                    } else {
22                        row++;
23                    }
24                } else {
25                    row++;
26                    col--;
27                }
28            } else {
29                if (row == 0 || col == width) {
30                    goingDown = true;
31                    if (col == width) {
32                        row++;
33                    } else {
34                        col++;
35                    }
36                } else {
37                    row--;
38                    col++;
39                }
40            }
41        }
42        return result;
43    }
44
45    public static boolean isOutOfBounds(int row, int col, int height, int width) {
46        return row < 0 || row > height || col < 0 || col > width;
47    }
48 }
49
```