[Diagonal Traverse - LeetCode](https://leetcode.com/problems/diagonal-traverse/solutions/459889/diagonal-traverse/)

[Diagonal Traverse - LeetCode](https://leetcode.com/problems/diagonal-traverse/solutions/2685483/diagonal-traverse-clear-solution-explanation-with-picture/?orderBy=hot)

O(n) Java Solution with simple logic

[shenlin11](https://leetcode.com/shenlin11/)

113

Aug 12, 2018

The basic idea here is that while changing the direction  
from moving up to moving down, the element on the right is always the first to traverse, whereas  
from moving down to moving up, the element below the current one is always the first to traverse.

class Solution {

public int[] findDiagonalOrder(int[][] matrix) {

if(matrix.length == 0) return new int[0];

int rows = matrix.length;

int columns = matrix[0].length;

int[] nums = new int[rows \* columns];

int currentRow = 0;

int currentColumn = 0;

boolean goUp = true;

for(int i = 0; i < nums.length; i++)

{

if(goUp) // moving up

{

nums[i] = matrix[currentRow--][currentColumn++];

// exceed the boundary

if(!(currentRow >= 0 && currentColumn <= columns - 1))

{

// return to the previous valid position

currentRow++;

currentColumn--;

// Going to the element right to it(same row, next column) is always preferable

// while changing the direction from moving up to moving down

// unless the next column is invalid

if(currentColumn < columns - 1) currentColumn++;

else currentRow++;

// will move down for next iteration

goUp = false;

}

}

else // moving down

{

nums[i] = matrix[currentRow++][currentColumn--];

// exceed the boundary

if(!(currentRow <= rows - 1 && currentColumn >= 0))

{

// return to the previous valid position

currentRow--;

currentColumn++;

// Going to the element below it(same column, next row) is always preferable

// while changing the direction from moving down to moving up

// unless the next row is invalid

if(currentRow < rows - 1) currentRow++;

else currentColumn++;

// will move up for next iteration

goUp = true;

}

}

}

return nums;

}

}