My Super Simple Solution. Can be used for both Spiral Matrix I and II

15.3K

**Works for any M\*N matrix!!!!!!**

This is my solution for Spiral Matrix I, <https://oj.leetcode.com/discuss/12228/super-simple-and-easy-to-understand-solution>. If you can understand that, this one is a no brainer :)

Guess what? I just made several lines of change (with comment "//change") from that and I have the following AC code:

public **class** **Solution** {

public int[][] generateMatrix(int n) {

// Declaration

int[][] matrix = **new** int[n][n];

// Normal Case

int rowStart = 0;

int rowEnd = n-1;

int colStart = 0;

int colEnd = n-1;

int num = 1; //change

**while** (rowStart <= rowEnd && colStart <= colEnd) {

**for** (int i = colStart; i <= colEnd; i ++) {

matrix[rowStart][i] = num ++; //change

}

rowStart ++;

**for** (int i = rowStart; i <= rowEnd; i ++) {

matrix[i][colEnd] = num ++; //change

}

colEnd --;

**for** (int i = colEnd; i >= colStart; i --) {

**if** (rowStart <= rowEnd)

matrix[rowEnd][i] = num ++; //change

}

rowEnd --;

**for** (int i = rowEnd; i >= rowStart; i --) {

**if** (colStart <= colEnd)

matrix[i][colStart] = num ++; //change

}

colStart ++;

}

**return** matrix;

}

}

Obviously, you could merge colStart and colEnd into rowStart and rowEnd because it is a square matrix. But this is easily extensible to matrices that are m\*n.