LeetCode 59

Yifeng Zeng

Description

59. Spiral Matrix II

Given an integer n, generate a square matrix filled with elements from 1 to n2 in spiral order.

For example,

Given n = 3,

You should return the following matrix:

[[1, 2, 3],

[8, 9, 4],

[7, 6, 5]]

Idea Report

This is a very straight forware step by step procedure.
We can have four integers to indicate the start and end point of each moving.
The top and left starts at 0, bottem and right starts at n - 1.

- First, we go from top left to top right, after the top row is done, we increase top boundary index by one (top++);
- Second, we go from top right to bottom right, after the rightmost column is done, we decrease right boundary index by one (right--);
- Third, we go from bottom right to bottom left, after the bottom row is done, we decrease

bottom boundary index by one (bottom--);

- Fourth, we go from bottom left to top left, after the leftmost column is done, we increase the left boundary index by one (left++);

Until the counter equals n * n, which means the matrix is fully filled, we return the matrix.

Code

```java
class Solution {
    public int[][] generateMatrix(int n) {
        int[][] res = new int[n][n];
        int count = 1;
        int top = 0;
        int bot = n - 1;
        int left = 0;
        int right = n - 1;

        while (count <= n * n) {
            for (int i = left; i <= right && count <= n * n; i++) {
                res[top][i] = count++;
            }
            top++;
            for (int i = top; i <= bot && count <= n * n; i++) {
                res[i][right] = count++;
            }
            right--;
            for (int i = right; i >= left && count <= n * n; i--) {
                res[bot][i] = count++;
            }
            bot--;

            for (int i = bot; i >= top && count <= n * n; i--) {
                res[i][left] = count++;
            }
            left++;
        }

        return res;
    }
}
```

Summary

- List all the possible steps and coding following the steps.