048. Rotate Image

048 Rotate Image

- array
- **Onion-rotate**

Description

You are given an *n* x *n* 2D matrix representing an image.

Rotate the image by 90 degrees (clockwise).

Note:

You have to rotate the image **in-place**, which means you have to modify the input 2D matrix directly. **DO NOT** allocate another 2D matrix and do the rotation.

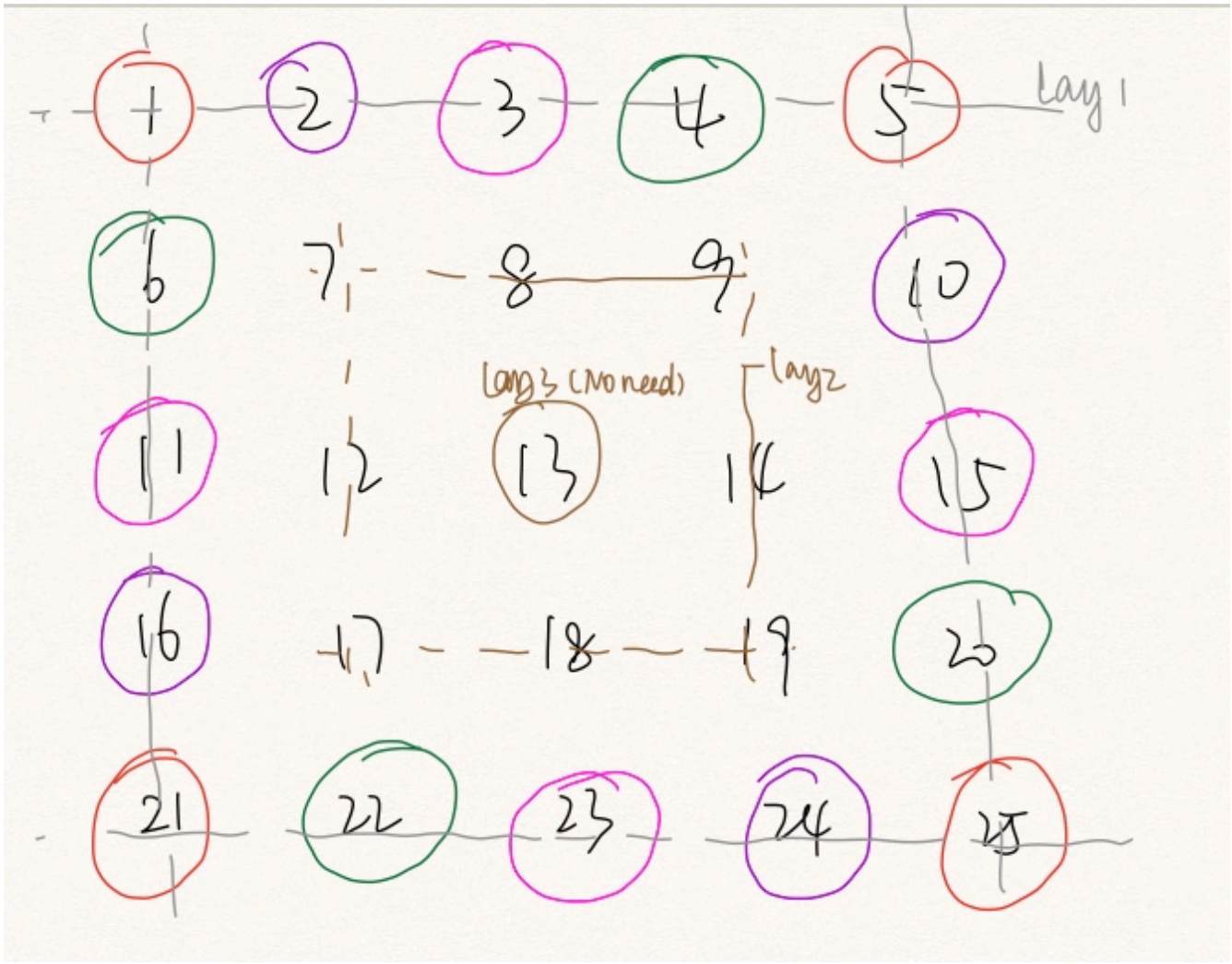
Example 1:

```
Given input matrix =
[
  [1,2,3],
  [4,5,6],
  [7,8,9]
],

rotate the input matrix in-place such that it becomes:
[
  [7,4,1],
  [8,5,2],
  [9,6,3]
]
```

Example 2:

```
Given input matrix =
[
  [ 5, 1, 9,11],
  [ 2, 4, 8,10],
  [13, 3, 6, 7],
  [15,14,12,16]
],

rotate the input matrix in-place such that it becomes:
[
  [15,13, 2, 5],
  [14, 3, 4, 1],
  [12, 6, 8, 9],
  [16, 7,10,11]
]
```

1. Thought line



2. Onion-rotate

```
class Solution {
public:
    void rotate(vector<vector<int> > &matrix) {
        int start = 0, end = matrix.size()-1;
        while(start<end) {
            for(int i=start; i<end; i++) {  // rotate a layer
                int offset = i - start;
                int temp = matrix[start][i];
                matrix[start][i] = matrix[end-offset][start];
                matrix[end-offset][start] = matrix[end][end-offset];
                matrix[end][end-offset] = matrix[start+offset][end];
                matrix[start+offset][end] = temp;
            }
            start++;
            end--;
        }
    }
};
```