LeetCode 48 – Rotate Image (Matrix Rotation)

Problem: Given an n x n matrix, rotate it by 90 degrees in-place. Rotation can be either Clockwise or Anti-Clockwise. In-place means without using any extra matrix.

Key Concept

Rotation is achieved using two steps: Transpose the matrix and then reverse rows or columns.

Clockwise Rotation (90°)

Steps: 1) Transpose the matrix 2) Reverse each row

Anti-Clockwise Rotation (90°)

Steps: 1) Transpose the matrix 2) Reverse each column

Time & Space Complexity

Time Complexity: $O(n^2)$ | Space Complexity: O(1)

C++ Code

```
// CLOCKWISE
void rotateClockwise(vector<vector<int>>& matrix) {
    int n = matrix.size();
    for(int i=0;i<n;i++)
        for(int j=i+1;j<n;j++)
            swap(matrix[i][j], matrix[j][i]);
    for(int i=0;i<n;i++)
        reverse(matrix[i].begin(), matrix[i].end());
}

// ANTICLOCKWISE
void rotateAntiClockwise(vector<vector<int>>& matrix) {
    int n = matrix.size();
    for(int i=0;i<n;i++)
        for(int j=i+1;j<n;j++)
            swap(matrix[i][j], matrix[j][i]);
    for(int j=0;j<n;j++){
        int top=0, bottom=n-1;
        while(top<bottom){
            swap(matrix[top][j], matrix[bottom][j]);
            top++; bottom--;
        }
    }
}
```

Java Code

```
// CLOCKWISE
public void rotateClockwise(int[][] matrix) {
    int n = matrix.length;
    for(int i=0;i<n;i++)
        for(int j=i+1;j<n;j++){
            int temp = matrix[i][j];
            matrix[i][j] = matrix[j][i];
            matrix[j][i] = temp;
        }
    for(int i=0;i<n;i++){
        int l=0, r=n-1;
        while(l<r){
            int temp = matrix[i][l];
            matrix[i][l] = matrix[i][r];
            matrix[i][r] = temp;
            l++; r--;
        }
    }
}

// ANTICLOCKWISE
public void rotateAntiClockwise(int[][] matrix) {
    int n = matrix.length;
    for(int i=0;i<n;i++)
        for(int j=i+1;j<n;j++){
            int temp = matrix[i][j];
            matrix[i][j] = matrix[j][i];
            matrix[j][i] = temp;
        }
    for(int j=0;j<n;j++){
        int top=0, bottom=n-1;
        while(top<bottom){
            int temp = matrix[top][j];
            matrix[top][j] = matrix[bottom][j];
            matrix[bottom][j] = temp;
            top++; bottom--;
        }
    }
}
```

Python Code

```
# CLOCKWISE
def rotate_clockwise(matrix):
    n = len(matrix)
    for i in range(n):
        for j in range(i+1, n):
            matrix[i][j], matrix[j][i] = matrix[j][i], matrix[i][j]
    for row in matrix:
        row.reverse()

# ANTICLOCKWISE
def rotate_anticlockwise(matrix):
    n = len(matrix)
    for i in range(n):
        for j in range(i+1, n):
            matrix[i][j], matrix[j][i] = matrix[j][i], matrix[i][j]
    for col in range(n):
        top, bottom = 0, n-1
        while top < bottom:
            matrix[top][col], matrix[bottom][col] = matrix[bottom][col], matrix[top][col]
            top += 1
            bottom -= 1
```