

Rotate Image (90 degrees clockwise) - Notes

The task is to rotate a given $n \times n$ matrix by 90 degrees (clockwise). ### Steps: 1. **Transpose the matrix**: Convert rows to columns by swapping $matrix[i][j]$ with $matrix[j][i]$. 2. **Reverse each row**: After transposition, reverse each row to complete the 90-degree clockwise rotation. ### Time Complexity: - $O(n^2)$, since we visit each element once during transpose and once during reversing. ### Space Complexity: - $O(1)$, as the rotation is done in-place without using extra space.

```
class Solution {
public:
    void rotate(vector<vector<int>>& matrix) {
        int n = matrix.size();
        int m = matrix[0].size();

        // Step 1: Transpose the matrix
        for(int i = 0; i < n - 1; i++){
            for(int j = i + 1; j < m; j++){
                swap(matrix[i][j], matrix[j][i]);
            }
        }

        // Step 2: Reverse each row
        for(int i = 0; i < n; i++){
            reverse(matrix[i].begin(), matrix[i].end());
        }
    }
};
```