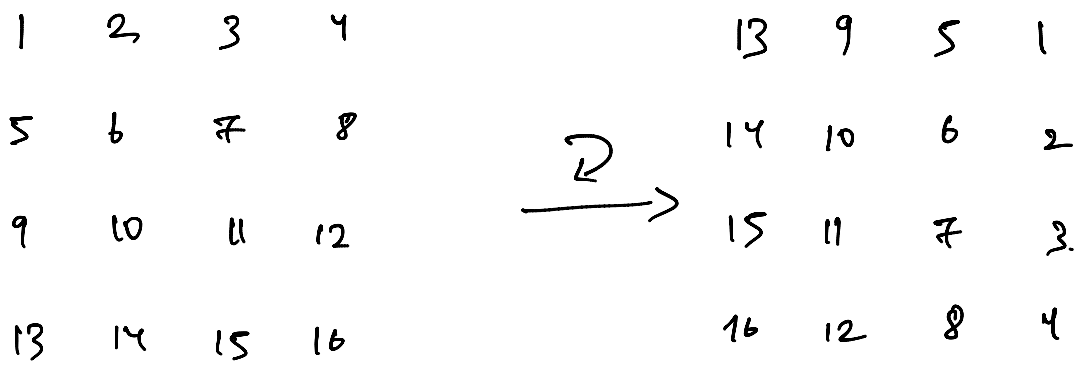
29. Rotate Matrix/Image

Rotate image by 90° (clockwise)



1) Brute force

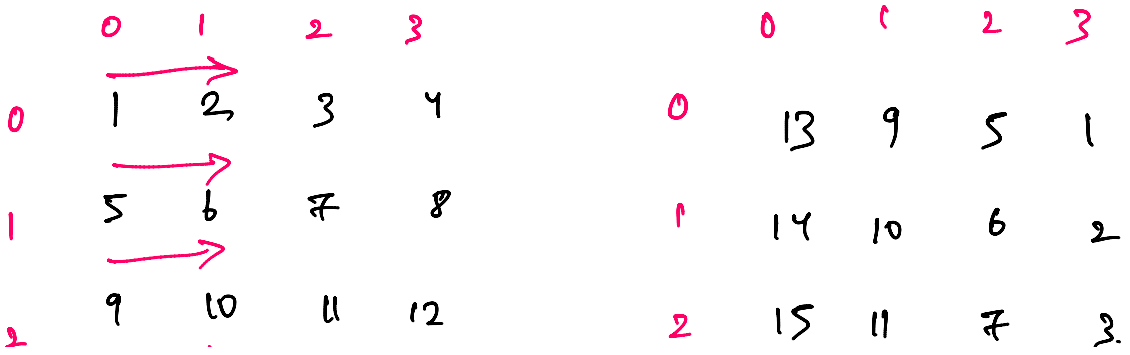
- Create a ans matrix of size (n x n)
- Place them at the correct place

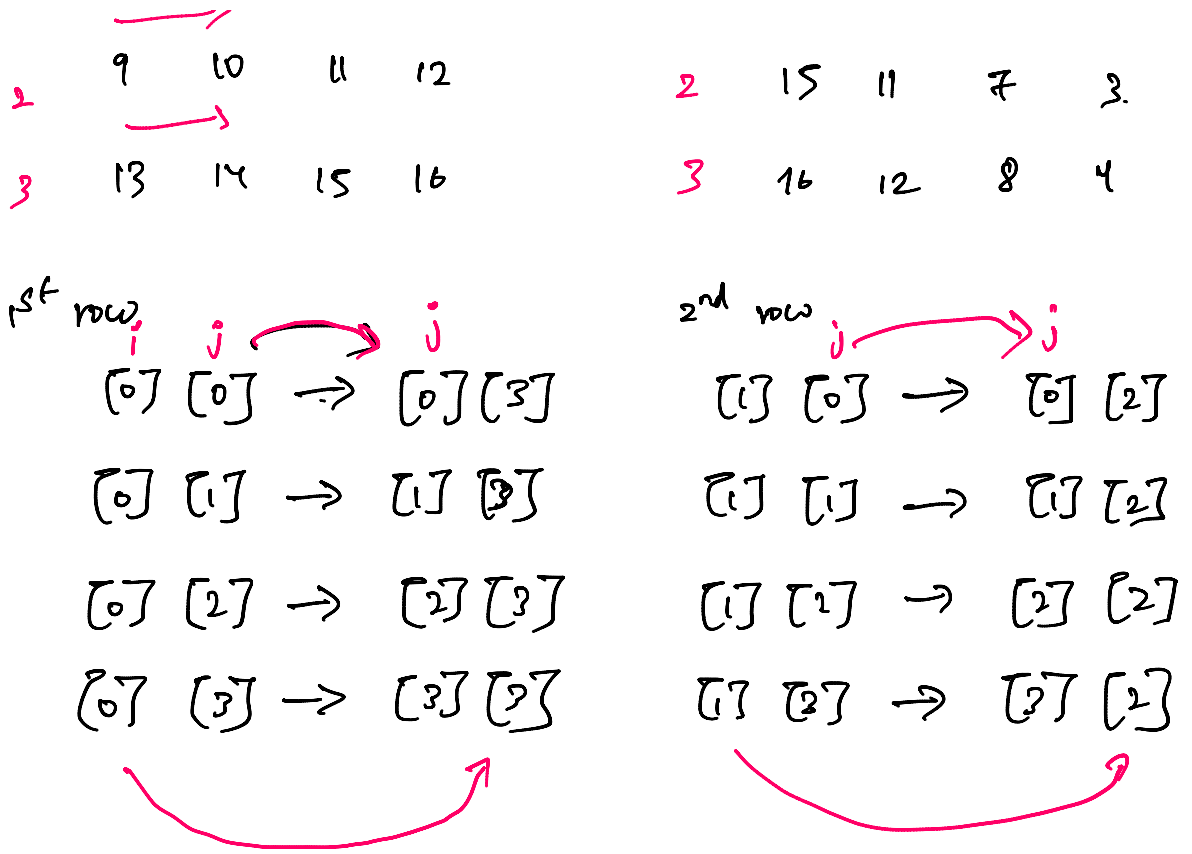
1st row → last column

2nd row → 2nd last

. .

. .






```
ans [n] [m]
for(i=0 → n)
{
    for(j=0 → n)
    {
        ans[j][n-i-1] = matrix[i][j]
```

}

}

T.C => $O(n^2)$ S.C => $O(n^2)$

2) 2nd Approach

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12
3	13	14	15	16

	0	1	2	3
0	13	9	5	1
1	14	10	6	2
2	15	11	7	3
3	16	12	8	4

- 1st column is in reverse order to the 1st row
- same for other column
- So we will do Transpose
  - row become column
  - column become row

1st step:
Transpose

1 5 9 13

2	6	10	14
3	7	11	15
4	8	12	16

2nd step
reverse every row.

13	9	5	1
14	10	6	2
15	11	7	3
16	12	8	4

} o/p

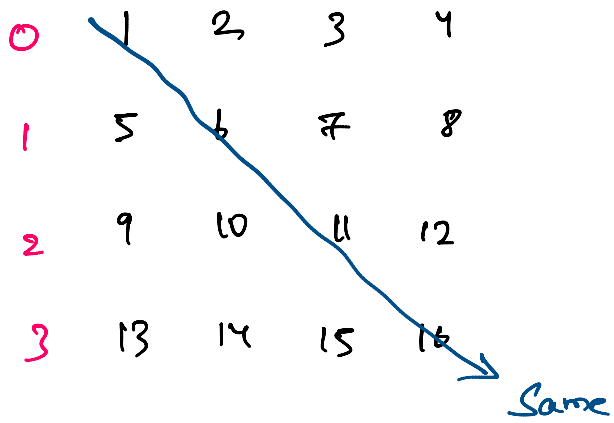
Steps:

1 → Transpose of Matrix

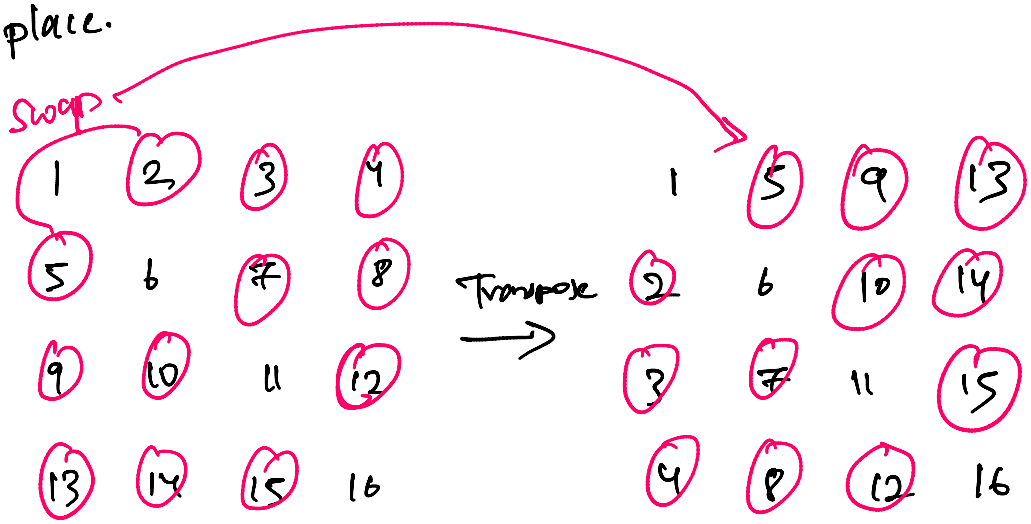
2 → Reversing every row.

→ All the diagonal of [0][0] [1][1] [2][2] [3][3] are staying same.

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12



- Now observe the things that are not staying at their place.



Swap

[0][1] → [1][0]
[0][2] → [2][0]
[0][3] → [3][0]
} 0th row

[1][2] → [2][1]
[1][3] → [3][1]
} 1st row

[2][3] → [3][2]
} 2nd row

- Just have to traverse for the right half

So for the 0th row, travel from 1 to 3

1st row, " " 2 to 3

2nd row, " " 3

0 ⟶ (1 to 3)

1 ⟶ (2 to 3)

i ⟶ (i+1 to n-1)

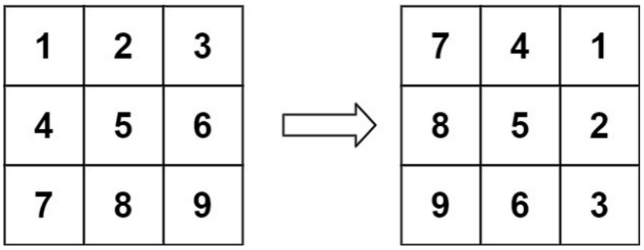
```
for(i=0 ⟶ n-2)
{
  for(j=i+1 ⟶ n-1)
  {
    swap(a[i][j], a[j][i])
  }
}
```

reverse it.

You are given an `n x n` 2D `matrix` representing an image, rotate the image by **90** degrees (clockwise).

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:




```
Input: matrix = [[1,2,3],[4,5,6],[7,8,9]]
Output: [[7,4,1],[8,5,2],[9,6,3]]
```

```cpp
class Solution {
public:
    void rotate(vector<vector<int>>& matrix) {
        int n = matrix.size();
        // 1st step => transpose
        for(int i = 0; i < n-1; i++){
            for(int j = i+1; j<n; j++){
                swap(matrix[i][j], matrix[j][i]);
            }
        }
        // 2nd step => reverse
        for(int i =0; i<n; i++){
            // row is mat[i]
            reverse(matrix[i].begin(), matrix[i].end());
        }
    }
};
```

T.C => O(n/2 * n/2) + O(n * n/2)

S.C => O(1)

```cpp
#include<vector>

void rotateMatrix(vector<vector<int>> &mat){
    int n = mat.size();
    // 1st step => transpose
    for(int i = 0; i < n-1; i++){
        for(int j = i+1; j<n; j++){
            swap(mat[i][j], mat[j][i]);
        }
    }
    // 2nd step => reverse
    for(int i =0; i<n; i++){
        // row is mat[i]
        reverse(mat[i].begin(), mat[i].end());
    }
}
```