

Rotate matrix elements clockwise

Easy Accuracy: 44.81% Submissions: 4K+ Points: 2



Don't Miss Out on the Chance to Work with Leading Companies! Visit the GFG Job Fair Now!

Given two integers M, N, and a 2D matrix Mat of dimensions MxN, clockwise rotate the elements in it.

Example 1:

Input:
 M=3,N=3
 Mat=[[1,2,3],[4,5,6],[7,8,9]]

Output:
 4 1 2
 7 5 3
 8 9 6

Explanation:
 Rotating the matrix clockwise gives this result.

Example 2:

C++ (g++ 5.4)

Average Time: 15m

Start Timer

```

1 // } Driver Code Ends
8 // User function Template for C++
9
10 class Solution {
11 public:
12     vector<vector<int>> rotateMatrix(int M, int N, vector<vector<int>> mat) {
13         // code here
14         int startRow = 0,endRow = M-1 , startCol = 0 , endCol = N-1;
15         while(startRow < endRow && startCol < endCol){
16             int prev = mat[startRow+1][startCol];
17
18             for(int i = startCol; i<=endCol ; i++){
19                 swap(prev,mat[startRow][i]);
20             }
21             startRow++;
22
23             for(int i = startRow ; i<=endRow ; i++){
24                 swap(prev, mat[i][endCol]);
25             }
26             endCol--;
27
28             for(int i = endCol ; i >= startCol ; i--){
29                 swap(prev, mat[endRow][i]);
30             }
31             endRow--;
32
33             for(int i = endRow ; i >= startRow; i--){
34                 swap(prev , mat[i][startCol]);
35             }
36             startCol++;
37         }
38         return mat;
39     }
40 };
41 // } Driver Code Ends

```

💡

Custom Input

Compile & Run

Submit



Maximum path sum in matrix



Medium

Accuracy: 42.59%

Submissions: 79K+

Points: 4



Don't Miss Out on the Chance to Work with Leading Companies! Visit the GFG Job Fair Now!



Given a $N \times N$ matrix of positive integers. There are only three possible moves from a cell $\text{Matrix}[r][c]$.

1. Matrix $[r+1][c]$
2. Matrix $[r+1][c-1]$
3. Matrix $[r+1][c+1]$

Starting from any column in row 0 return the largest sum of any of the paths up to row $N-1$.

NOTE: We can start from any column in zeroth row and can end at any column in $(N-1)$ th row.

Example 1:

Input: $N = 2$

Matrix = $\{\{348, 391\},$
 $\{618, 193\}\}$

Output: 1009

```
1 // } Driver Code Ends
2 // User function Template for C++
3
4 class Solution{
5 public:
6     int maximumPath(int n, vector<vector<int>> Matrix)
7     {
8         // code here
9         for(int i = n-2 ; i >= 0 ; i--){
10             for(int j = 0 ; j < n ; j++){
11
12                 int a = (j-1)>=0 ? Matrix[i+1][j-1] : 0;
13                 int b = (j+1)<n ? Matrix[i+1][j+1] : 0;
14                 int c = Matrix[i+1][j];
15
16                 Matrix[i][j] = Matrix[i][j] + max(a,max(b,c));
17             }
18         }
19
20         int res = *max_element(begin(Matrix[0]), end(Matrix[0]));
21
22         return res;
23     }
24 };
25 // } Driver Code Ends
```

