54. Spiral Matrix

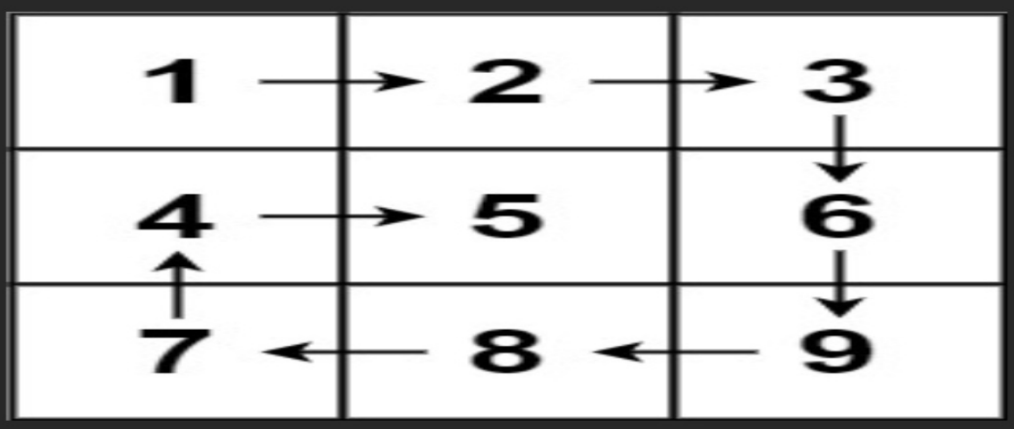
Hint

Medium 12.6K 1.1K

Companies

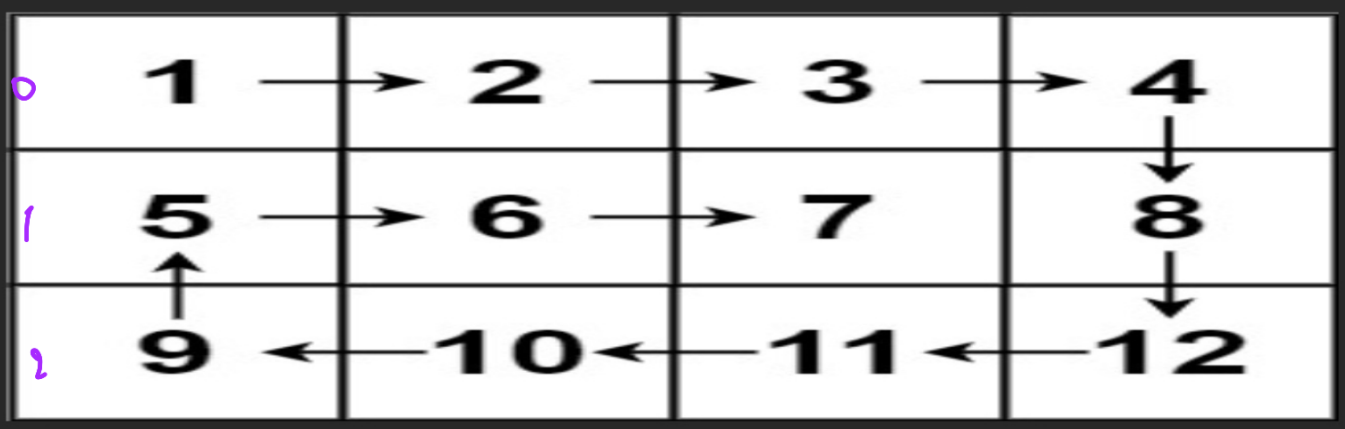
Given an `m x n` `matrix`, return *all elements of the* `matrix` *in spiral order.*

Example 1:




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

Example 2:

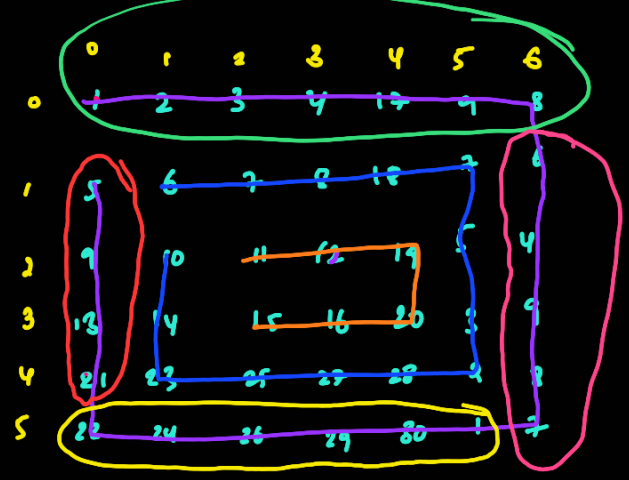



```
Input: matrix = [[1,2,3,4],[5,6,7,8],
[9,10,11,12]]
Output: [1,2,3,4,8,12,11,10,9,5,6,7]
```

Constraints:

- `m == matrix.length`
- `n == matrix[i].length`
- `1 <= m, n <= 10`
- `-100 <= matrix[i][j] <= 100`

Accepted **1.1M** | Submissions **2.4M** |
Acceptance Rate **46.7%**



This problem requires just keen observation and requires us to find pattern using many variables

```
class Solution {
public:
    vector<int> spiralOrder(vector<vector<int>>& matrix) {
        vector<int> ans;
        int x=matrix[0].size();
        int y=matrix.size();
        int n=x*y;
        int r=0,c=0;
        int i=0,j=0;
        int t= ceil(y/2.0);
        while(r < t)
        {
            i=r,j=c;
            while(j<x && n>0) ans.push_back(matrix[r][j++]),n--;
            while(i<(y-1) && n>0) ans.push_back(matrix[++i][x-1]),n--;
            j=j-2;
            while(j>=c && n>0) ans.push_back(matrix[y-1][j--]),n--;
            i--;
            while(i>r && n>0) ans.push_back(matrix[i--][c]),n--;
            r++;
            c++;
            x--,y--;
        }

        return ans;
    }
};
```

we are keeping track of this just to make sure that no element gets repeated again.

The logic is absolutely working fine but for some matrices it is printing some elements twice. So To avoid that we are using a n variable.

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix}$$

This is getting printed twice if we don't use n variable