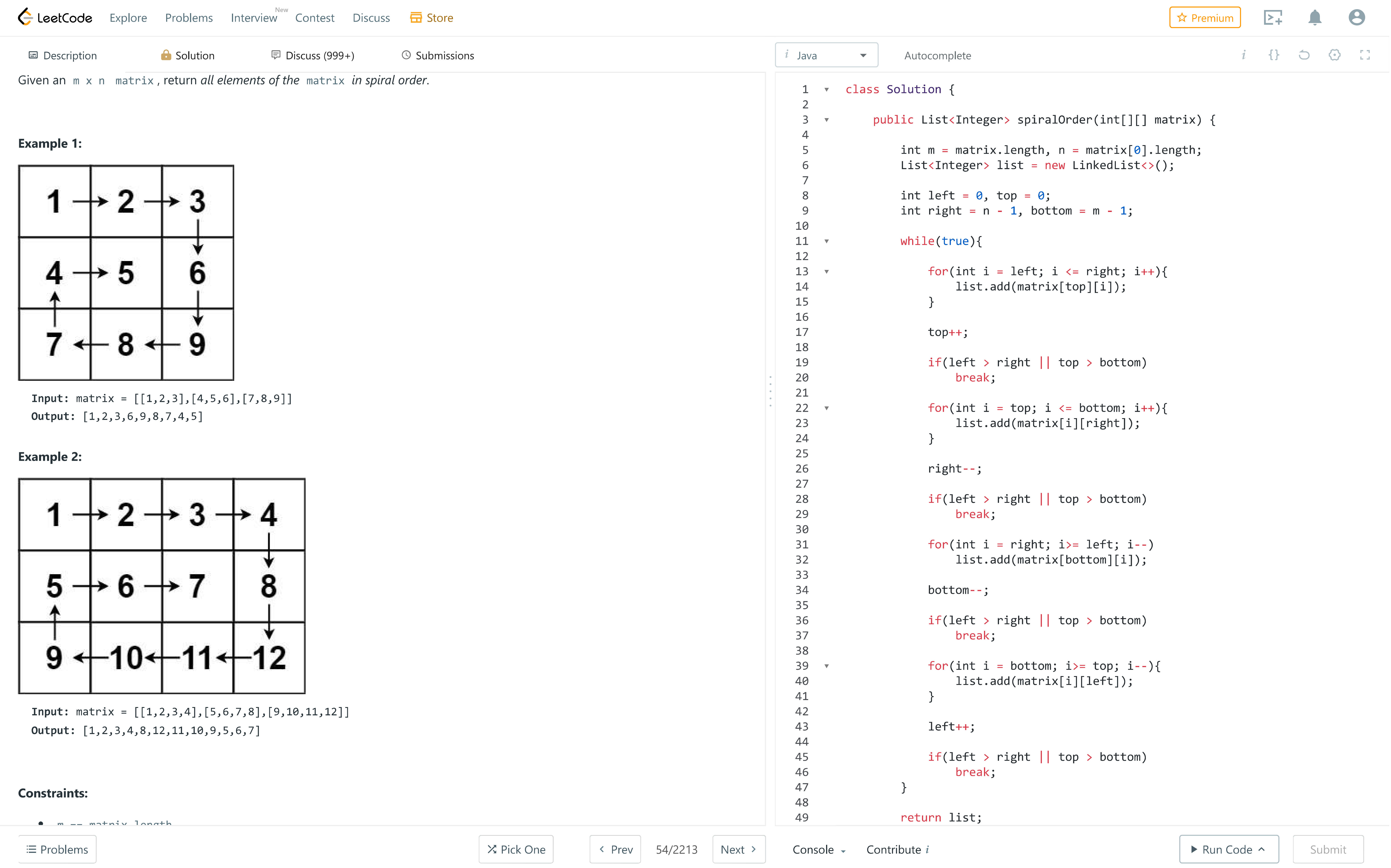

Given an `m x n` `matrix`, return *all elements of the* `matrix` *in spiral order*.

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`

```java
class Solution {

    public List<Integer> spiralOrder(int[][] matrix) {

        int m = matrix.length, n = matrix[0].length;
        List<Integer> list = new LinkedList<>();

        int left = 0, top = 0;
        int right = n - 1, bottom = m - 1;

        while(true){

            for(int i = left; i <= right; i++){
                list.add(matrix[top][i]);
            }

            top++;

            if(left > right || top > bottom)
                break;

            for(int i = top; i <= bottom; i++){
                list.add(matrix[i][right]);
            }

            right--;

            if(left > right || top > bottom)
                break;

            for(int i = right; i>= left; i--)
                list.add(matrix[bottom][i]);

            bottom--;

            if(left > right || top > bottom)
                break;

            for(int i = bottom; i>= top; i--){
                list.add(matrix[i][left]);
            }

            left++;

            if(left > right || top > bottom)
                break;
        }

        return list;
```