面试题29：顺时针打印矩阵

题目描述

输入一个矩阵，按照从外向里以顺时针的顺序依次打印出每一个数字。

输入描述

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

则依次打印出数字 1,2,3,4,8,12,16,15,14,13,9,5,6,7,11,10。

解题思路

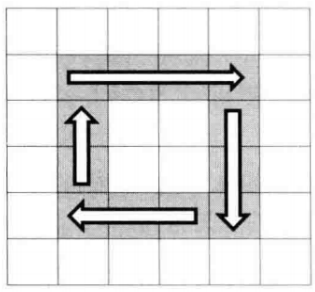

图 4.4　把矩阵看成由若干个顺时针方向的圈组成

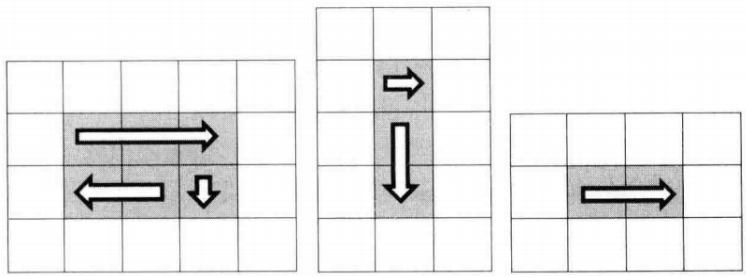

图 4.5　打印矩阵最里面一圈可能只需要三步、两步甚至一步

C++

```cpp
/*思想，用左上和右下的坐标定位出一次要旋转打印的数据，一次旋转打印结束后，往对角分别前进和后退一个单位。提交代码时，主要的问题出在没有控制好后两个for循环，需要加入条件判断，防止出现单行或者单列的情况。
 */
class Solution {
public:
    vector<int> printMatrix(vector<vector<int> > matrix) {
        int row = matrix.size();
        int col = matrix[0].size();
        vector<int> res;
        // 输入的二维数组非法，返回空的数组
        if (row == 0 || col == 0)  return res;
        // 定义四个关键变量，表示左上和右下的打印范围
        int left = 0, top = 0, right = col - 1, bottom = row - 1;
        while (left <= right && top <= bottom){
            // left to right
            for (int i = left; i <= right; ++i)  res.push_back(matrix[top][i]);
            // top to bottom
            for (int i = top + 1; i <= bottom; ++i)  res.push_back(matrix[i][right]);
            // right to left
            if (top != bottom)
```

```
            for (int i = right - 1; i >= left; --i)  res.push_back(matrix[bottom][i]);
            // bottom to top
            if (left != right)
            for (int i = bottom - 1; i > top; --i)  res.push_back(matrix[i][left]);
            left++,top++,right--,bottom--;
        }
        return res;
    }
};
```

C++

```
/*解题思路：顺时针打印就是按圈数循环打印，一圈包含两行或者两列，在打印的时候会出现某一圈中只包含一行，要判断从左向右打印和从右向左打印的时候是否会出现重复打印，同样只包含一列时，要判断从上向下打印和从下向上打印的时候是否会出现重复打印的情况*/
class Solution {
public:
    vector<int> printMatrix(vector<vector<int> > matrix) {
        vector<int>res;
        res.clear();
        int row=matrix.size();//行数
```

```
        int collor=matrix[0].size();//列数
        //计算打印的圈数
        int circle=((row<collor?
row:collor)-1)/2+1;//圈数
        for(int i=0;i<circle;i++){
            //从左向右打印
            for(int j=i;j<collor-i;j++)
                res.push_back(matrix[i][j]);

            //从上往下的每一列数据
            for(int k=i+1;k<row-i;k++)
                res.push_back(matrix[k][collor-
1-i]);
            //判断是否会重复打印(从右向左的每行数据)
            for(int m=collor-i-2;(m>=i)&&(row-i-
1!=i);m--)
                res.push_back(matrix[row-i-1]
[m]);
            //判断是否会重复打印(从下往上的每一列数
据)
            for(int n=row-i-2;(n>i)&&(collor-i-
1!=i);n--)
                res.push_back(matrix[n][i]);}
        return res;
    }
};
```

Python

```python
# -*- coding:utf-8 -*-
class Solution:
    # matrix类型为二维列表，需要返回列表
    def printMatrix(self, matrix):
        # write code here
        res = []
        while matrix:
            res += matrix.pop(0)
            if matrix and matrix[0]:
                for row in matrix:
                    res.append(row.pop())
            if matrix:
                res += matrix.pop()[::-1]
            if matrix and matrix[0]:
                for row in matrix[::-1]:
                    res.append(row.pop(0))
        return res
```

Java

```java
import java.util.ArrayList;
public class Solution {
    public ArrayList<Integer> printMatrix(int [][] matrix) {
        ArrayList<Integer> result = new ArrayList<Integer> ();
        if(matrix.length==0) return result;
```

```
6           int n = matrix.length,m =
matrix[0].length;
7           if(m==0) return result;
8           //这个是层数
9           int layers = (Math.min(n,m)-1)/2+1;
10          for(int i=0;i<layers;i++){
11              //左至右
12              for(int k = i;k<m-i;k++)
result.add(matrix[i][k]);
13              //右上至右下
14              for(int j=i+1;j<n-i;j++)
result.add(matrix[j][m-i-1]);
15              //右至左
16              for(int k=m-i-2;(k>=i)&&(n-i-
1!=i);k--) result.add(matrix[n-i-1][k]);
17              //左下至左上
18              for(int j=n-i-2;(j>i)&&(m-i-1!=i);j-
-) result.add(matrix[j][i]);
19          }
20          return result;
21      }
22  }
```