

第43题——顺时针打印矩阵

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。

示例 1:

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

```
输出: [1,2,3,6,9,8,7,4,5]
```

示例 2:

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

```
输出: [1,2,3,4,8,12,11,10,9,5,6,7]
```

限制:

$0 \leq \text{matrix.length} \leq 100$ $0 \leq \text{matrix}[i].\text{length} \leq 100$

模拟(1)

先定义好边界:

```
1. left=0;
```

```
2. right=matrix[0].size()-1;
```

```
3. top=0;
```

```
4. below=matrix.size()-1;
```

然后按照逆时针顺序模拟,把数值放进vector容器中

1. 从左到右

2. 从上到下

3. 从右到左

4. 从下到上

```
class Solution {
public:
    vector<int> spiralOrder(vector<vector<int>>& matrix) {
        vector<int> res;
        if(matrix.size()==0 || matrix[0].size()==0)
        {
            return res;
        }
        int t=0;//上
        int l=0;//左
        int r=matrix[0].size();//右
        int b=matrix.size();//下

        while(1)
        {
            //从左往右
            for(int i=l;i<r;i++){
                res.push_back(matrix[t][i]);
            }
            t++;
            if(t>b)break;

            //从上到下
            for(int i=t;i<b;i++){
                res.push_back(matrix[i][r]);
            }
            r--;
            if(r<l)break;

            //从右向左
            for(int i=r;i>l;i--){
                res.push_back(matrix[b][i]);
            }
            b--;
            if(b<t)break;

            //从下到上
            for(int i=b;i>t;i--){
```

```
        res.push_back(matrix[i][l]);
    }
    l++;
    if(l>r)break;
}
return res;
}
};
```