常规思路（这个思路应该是比较清晰的）

```cpp
class Solution {
public:
    vector<int> spiralOrder(vector<vector<int>>& matrix) {
      vector<int> result;
        if (matrix.empty()) return result; // 处理空矩阵

        int M = matrix.size();
        int N = matrix[0].size();

        int top = 0, bottom = M - 1;
        int left = 0, right = N - 1;

        while (top ≤ bottom && left ≤ right) {
            // 从左到右遍历当前上边界
            for (int j = left; j ≤ right; j++) {
                result.push_back(matrix[top][j]);
            }

            // 上边界下移
            top++;

            // 从上到下遍历当前右边界
            for (int i = top; i ≤ bottom; i++) {
                result.push_back(matrix[i][right]);
            }

            // 右边界左移
            right--;

            if (top ≤ bottom) {
                // 从右到左遍历当前下边界
                for (int j = right; j ≥ left; j--) {
                    result.push_back(matrix[bottom][j]);
                }
                // 下边界上移
                bottom--;
            }

            if (left ≤ right) {
                // 从下到上遍历当前左边界
                for (int i = bottom; i ≥ top; i--) {

```

```
                    result.push_back(matrix[i][left]);
                }
                // 左边界右移
                left++;
            }
        }

        return result;
    }
};
```