剑指29 顺时针打印矩阵

纯数组/二维数组的边界判定。 但是不能算简单题.....

限制为：

- 0 <= matrix.length <= 100
- 0 <= matrix[i].length <= 100

所以0,0的时候matrix[0].size()会出错！！！！
必须提前判断空！！！！

```
// 我的答案
// 经提示才改为matrix为空的时候return {};

class Solution {
public:
    vector<int> spiralOrder(vector<vector<int>>& matrix) {
        if (matrix.size() == 0 || matrix[0].size() == 0) {
            return {};
        }
        int right = matrix[0].size();
        int down = matrix.size() ;
        int up = -1;
        int left = -1;
        int x,y = 0;
        int n = 0;
        vector<int> out;
        while (left+1 < right && up+1 < down) {
            switch (n % 4) {
                case 0 : {
                    x = up + 1;
                    for (y = left + 1; y < right; y++) {
                        out.push_back(matrix[x][y]);
                    }
                    up++;
                    break;
                }
                case 1 : {
                    y = right - 1;
                    for (x = up + 1; x < down; x++) {
                        out.push_back(matrix[x][y]);
                    }
                    right--;
                    break;
                }
                case 2 : {
                    x = down - 1;
                    for (y = right - 1; y > left; y--) {
                        out.push_back(matrix[x][y]);
                    }
```

```
                down--;
                break;
            }
            case 3 : {
                y = left + 1;
                for (x = down - 1; x > up; x--) {
                    out.push_back(matrix[x][y]);
                }
                left++;
                break;
            }
        }
        n++;
    }
    return out;
  }
};

// 虽然但是其实我这个是状态机思路，可以改n++为每个case后转状态。
```