# 题目

给你一个 m \* n 的矩阵，矩阵中的数字 各不相同 。请你按 任意 顺序返回矩阵中的所有幸运数。

幸运数是指矩阵中满足同时下列两个条件的元素：

在同一行的所有元素中最小

在同一列的所有元素中最大

示例 1：

输入：matrix = [[3,7,8],[9,11,13],[15,16,17]]

输出：[15]

解释：15 是唯一的幸运数，因为它是其所在行中的最小值，也是所在列中的最大值。

示例 2：

输入：matrix = [[1,10,4,2],[9,3,8,7],[15,16,17,12]]

输出：[12]

解释：12 是唯一的幸运数，因为它是其所在行中的最小值，也是所在列中的最大值。

示例 3：

输入：matrix = [[7,8],[1,2]]

输出：[7]

解释：7是唯一的幸运数字，因为它是行中的最小值，列中的最大值。

提示：

m == mat.length

n == mat[i].length

1 <= n, m <= 50

1 <= matrix[i][j] <= 10^5

矩阵中的所有元素都是不同的

# 分析

## 方法一：模拟

思路：

代码：

class Solution {

public:

vector<int> luckyNumbers (vector<vector<int>>& matrix) {

int m = matrix.size();

int n = matrix[0].size();

vector<int> rowMin(m, INT\_MAX);

vector<int> colMax(n, INT\_MIN);

for (int i = 0; i < m; ++i) {

for (int j = 0; j < n; ++j) {

rowMin[i] = min(rowMin[i], matrix[i][j]);

colMax[j] = max(colMax[j], matrix[i][j]);

}

}

vector<int> luckyNumbers;

for (int i = 0; i < m; ++i) {

for (int j = 0; j < n; ++j) {

if (matrix[i][j] == rowMin[i] && matrix[i][j] == colMax[j]) {

luckyNumbers.push\_back(matrix[i][j]);

}

}

}

return luckyNumbers;

}

};