Problem 1380: Lucky Numbers in a Matrix

Problem Information

Difficulty: Easy
Acceptance Rate: 79.96%
Paid Only: No
Tags: Array, Matrix

Problem Description

Given an `m x n` matrix of **distinct** numbers, return _all**lucky numbers** in the matrix in **any** order_.

A **lucky number** is an element of the matrix such that it is the minimum element in its row and maximum in its column.

Example 1:

Input: matrix = [[3,7,8],[9,11,13],[15,16,17]] **Output:** [15] **Explanation:** 15 is the only lucky number since it is the minimum in its row and the maximum in its column.

Example 2:

Input: matrix = [[1,10,4,2],[9,3,8,7],[15,16,17,12]] **Output:** [12] **Explanation:** 12 is the only lucky number since it is the minimum in its row and the maximum in its column.

Example 3:

Input: matrix = [[7,8],[1,2]] **Output:** [7] **Explanation:** 7 is the only lucky number since it is the minimum in its row and the maximum in its column.

Constraints:

* `m == mat.length` * `n == mat[i].length` * `1 <= n, m <= 50` * `1 <= matrix[i][j] <= 105`. * All elements in the matrix are distinct.

Code Snippets

C++:

```
class Solution {
public:
vector<int> luckyNumbers(vector<vector<int>>& matrix) {

}
};
```

Java:

```
class Solution {
public List<Integer> luckyNumbers(int[][] matrix) {

}
}
```

Python3:

```
class Solution:
def luckyNumbers(self, matrix: List[List[int]]) -> List[int]:
```