

矩阵中的幸运数

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]

```
class Solution:
    def luckyNumbers (self, matrix: List[List[int]]) -> List[int]:
        minset = set()
        maxset = set()
        row= len(matrix)
        col=len(matrix[0])

        for h in range(row): # 取出同一行最小
            minset.add(min(matrix[h]))

        for k in range(col): # 取出同一列最大
            maxset.add(max(x[k] for x in matrix))

        # 求二者交集，并且转换输出类型
        return list(minset.intersection(maxset))
```