Problem 2282: Number of People That Can Be Seen in a Grid

Problem Information

Difficulty: Medium
Acceptance Rate: 47.49%
Paid Only: Yes
Tags: Array, Stack, Matrix, Monotonic Stack

Problem Description

You are given an `m x n` **0-indexed** 2D array of positive integers `heights` where `heights[i][j]` is the height of the person standing at position `(i, j)`.

A person standing at position `(row1, col1)` can see a person standing at position `(row2, col2)` if:

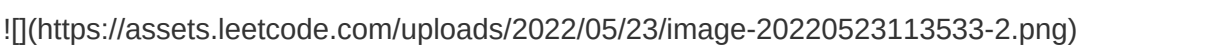
* The person at `(row2, col2)` is to the right **or** below the person at `(row1, col1)`. More formally, this means that either `row1 == row2` and `col1 < col2` **or** `row1 < row2` and `col1 == col2`. * Everyone in between them is shorter than **both** of them.

Return _an_`m x n` _2D array of integers_`answer` _where_`answer[i][j]`_is the number of people that the person at position_`(i, j)`_can see._

Example 1:

Input: heights = [[3,1,4,2,5]] **Output:** [[2,1,2,1,0]] **Explanation:** - The person at (0, 0) can see the people at (0, 1) and (0, 2). Note that he cannot see the person at (0, 4) because the person at (0, 2) is taller than him. - The person at (0, 1) can see the person at (0, 2). - The person at (0, 2) can see the people at (0, 3) and (0, 4). - The person at (0, 3) can see the person at (0, 4). - The person at (0, 4) cannot see anybody.

Example 2:



Input: heights = [[5,1],[3,1],[4,1]] **Output:** [[3,1],[2,1],[1,0]] **Explanation:** - The person at (0, 0) can see the people at (0, 1), (1, 0) and (2, 0). - The person at (0, 1) can see the person at (1, 1). - The person at (1, 0) can see the people at (1, 1) and (2, 0). - The person at (1, 1) can see the person at (2, 1). - The person at (2, 0) can see the person at (2, 1). - The person at (2, 1) cannot see anybody.

Constraints:

* `1 <= heights.length <= 400` * `1 <= heights[i].length <= 400` * `1 <= heights[i][j] <= 105`

Code Snippets

C++:

```
class Solution {
public:
vector<vector<int>> seePeople(vector<vector<int>>& heights) {

}
};
```

Java:

```
class Solution {
public int[][] seePeople(int[][] heights) {

}
}
```

Python3:

```
class Solution:
def seePeople(self, heights: List[List[int]]) -> List[List[int]]:
```