

803. Bricks Falling When Hit

Description

You are given an `m x n` binary `grid`, where each `1` represents a brick and `0` represents an empty space. A brick is **stable** if:

- It is directly connected to the top of the grid, or
- At least one other brick in its four adjacent cells is **stable**.

You are also given an array `hits`, which is a sequence of erasures we want to apply. Each time we want to erase the brick at the location `hits[i] = (rowi, coli)`. The brick on that location (if it exists) will disappear. Some other bricks may no longer be stable because of that erasure and will **fall**. Once a brick falls, it is **immediately** erased from the `grid` (i.e., it does not land on other stable bricks).

Return *an array `result`, where each `result[i]` is the number of bricks that will **fall** after the `ith` erasure is applied.*

Note that an erasure may refer to a location with no brick, and if it does, no bricks drop.

Example 1:

```
Input: grid = [[1,0,0,0],[1,1,1,0]], hits = [[1,0]]
Output: [2]
Explanation: Starting with the grid:
[[1,0,0,0],
 [1,1,1,0]]
We erase the underlined brick at (1,0), resulting in the grid:
[[1,0,0,0],
 [0,1,1,0]]
The two underlined bricks are no longer stable as they are no longer connected to the top nor adjacent to another stable brick, so they will fall.
The resulting grid is:
[[1,0,0,0],
 [0,0,0,0]]
Hence the result is [2].
```

Example 2:

```
Input: grid = [[1,0,0,0],[1,1,0,0]], hits = [[1,1],[1,0]]
Output: [0,0]
Explanation: Starting with the grid:
[[1,0,0,0],
 [1,1,0,0]]
We erase the underlined brick at (1,1), resulting in the grid:
[[1,0,0,0],
 [1,0,0,0]]
All remaining bricks are still stable, so no bricks fall. The grid remains the same:
[[1,0,0,0],
 [1,0,0,0]]
Next, we erase the underlined brick at (1,0), resulting in the grid:
[[1,0,0,0],
 [0,0,0,0]]
Once again, all remaining bricks are still stable, so no bricks fall.
Hence the result is [0,0].
```

Constraints:

- `m == grid.length`
- `n == grid[i].length`
- `1 <= m, n <= 200`
- `grid[i][j]` is `0` or `1`.
- `1 <= hits.length <= 4 * 104`
- `hits[i].length == 2`
- `0 <= xi <= m - 1`
- `0 <= yi <= n - 1`
- All `(xi, yi)` are unique.

