[CS 11 25.1] Lab 4g – Spy Satellite

Cheatsheet is available here: https://oj.dcs.upd.edu.ph/cs11cheatsheet/

Submit solution [CS 11 25.1] Lab Exercise 4

✓ **Points:** 200 (partial)

Time limit: 4.0s

Memory limit: 1G

› **Problem type**

⌄ **Allowed languages**
py3

Problem Statement

Using a spying software called Strava, you have been able to track the movement of a particular soldier in a secret military base.

The area of the base the soldier is patrolling can be represented as an $r \times c$ grid. We number the rows 0 to $r-1$ from top to bottom, and 0 to $c-1$ from left to right. We denote the cell at row i and column j by (i, j).

You know that the soldier's routine consists of walking around in this grid as follows:

- First, start at cell (i_0, j_0).
- Next, go to cell (i_1, j_1).
- Next, go to cell (i_2, j_2).
- ...
- Finally, go to cell (i_n, j_n).

Also, to go from some cell (i_k, j_k) to the next cell (i_{k+1}, j_{k+1}) in the list, you know that the soldier takes the shortest path, and also moves *horizontally* first before vertically. For example, to go from $(3, 1)$ to $(1, 4)$, the following sequence of cells are visited:

$$(3,1) \to (3,2) \to (3,3) \to (3,4) \to (2,4) \to (1,4).$$

Given the sequence of cells visited, please determine the number of times the soldier visits every cell in the grid.

Task Details

Your task is to implement a function called `patrol_heatmap`. The function takes three argument:

- the first is the `int` r.
- the second is the `int` c.
- the third is a a `tuple` of $n+1$ pairs of `int`s (i, j) representing the cells the soldier visits.

The function must return a grid represented as a `list` of r `list`s, each of which is a `list` of c `int`s, representing the number of visits the soldier makes on each cell over the course of the routine.

Restrictions

(See 4a for more restrictions)

For this problem:

- Loops and lists are allowed.
- Up to 8 function definitions are allowed.
- Recursion is **disallowed**. (The recursion limit has been greatly reduced.)
- Sets and dictionaries are allowed.
- Generators and comprehensions are allowed.
- The source code limit is $3{,}000$.

Example Calls

Example 1 Function Call

Copy

```
patrol_heatmap(5, 6, (
    (3, 4),
    (3, 1),
    (1, 4),
    (4, 1),
    (1, 5),
    (0, 3),
))
```

Example 1 Return Value

Copy

```
[
    [0, 0, 0, 1, 0, 0],
    [0, 1, 1, 2, 2, 1],
    [0, 1, 0, 0, 1, 1],
    [0, 2, 2, 2, 2, 1],
    [0, 1, 1, 1, 1, 1],
]
```

Example 1 Explanation

The sequence of cells visited are:

$$(3,4) \to (3,3) \to (3,2) \to (3,1) \to (3,2) \to (3,3) \to (3,4) \to$$
$$(2,4) \to (1,4) \to (1,3) \to (1,2) \to (1,1) \to (2,1) \to (3,1) \to$$
$$(4,1) \to (4,2) \to (4,3) \to (4,4) \to (4,5) \to (3,5) \to (2,5) \to$$
$$(1,5) \to (1,4) \to (1,3) \to (0,3)$$

Example 2 Function Call

Copy

```
patrol_heatmap(5, 6, (
    (0, 0),
    (0, 0),
    (0, 0),
))
```

Example 2 Return Value

Copy

```
[
    [1, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0],
]
```

Constraints

- The function `patrol_heatmap` will be called at most $50{,}000$ times.
- $1 \le r, c, rc \le 150{,}000$.
- $1 \le n \le 150{,}000$.
- The sum of the rc is $\le 150{,}000$.
- The sum of the n is $\le 150{,}000$.
- $0 \le i_k < r$
- $0 \le j_k < c$

Scoring

Note: New tests may be added and all submissions may be rejudged at a later time. (All future tests will satisfy the constraints.)

- You get 100 ● points if you solve all test cases where:
  - $rc \le 50$
  - $n \le 50$
  - the sum of all n is ≤ 500.
- You get 25 ● points if you solve all test cases where:
  - $rc \le 50$
  - $n \le 4{,}000$
  - the sum of all n is $\le 8{,}000$.
- You get 25 ● points if you solve all test cases where:
  - $rc \le 50$
- You get 50 💔 points if you solve all test cases.

Clarifications

Report an issue

No clarifications have been made at this time.