[CS 11 25.1] HOPE 1 B3 – Gym Challenge

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

Submit solution [CS 11 25.1] HOPE 1

Points: 175 (partial)
Time limit: 12.0s
Memory limit: 2G

Problem type

Allowed languages
py3

Problem Statement

You're on your way towards getting your next gym badge! You enter the gym, and...oh, what do we have here?

The gym contains an $r \times c$ grid of trash cans. Yes, trash cans.

The rows in this grid are numbered 0 to $r - 1$ from top to bottom, and the columns are numbered 0 to $c - 1$ from left to right. The trash can at the ith row and jth column has *location* (i, j).

Apparently, two of these trash cans contain a button that you should press, and these two trash cans are guaranteed to be *orthogonally adjacent*. By orthogonally adjacent, consider the following:

```
XXXXX
XX0XX
X0A0X
XX0XX
XXXXX
```

Here, the cells marked `0` are orthogonally adjacent to `A`, while the cells marked `X` aren't.

In other words, two cells are orthogonally adjacent if they share an edge.

What are all possible locations of the two buttons?

> **Fun fact:** This is the actual gimmick used for one of the gyms in one of the Pokémon games! Do you know which gym it is?

Task Details

Your task is to implement a function named `poss_locations`. This function has two parameters: r and c. Both of these are integers.

The function must return a `frozenset` of `frozenset`s. Each frozenset should contain exactly two pairs denoting the possible locations of the two buttons.

Restrictions

- Recursion is **disallowed**.
- Your source code must have at most 800 bytes.

Examples

Example 1 Function Call

```
poss_locations(1, 2)
```

Example 1 Return Value

```
frozenset((
    frozenset(((0, 0), (0, 1))),
))
```

Constraints

- The function `poss_locations` will be called at most $70{,}000$ times.
- $1 \leq r, c$
- The sum of rc across all test cases is at most $350{,}000$.

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 50 ❤️ points if you solve all test cases where:
  - $r = 1$ **or** $c = 1$.
- You get 100 🔴 points if you solve all test cases where:
  - The sum of rc across all test cases is at most 300.
- You get 25 🔴 points if you solve all test cases.

Clarifications

Report an issue

No clarifications have been made at this time.