



[CS 11 25.1] Lab 2c – Koyuki and Bombs 2

Problem Statement

[Submit solution](#)

Koyuki is testing out a new kind of bomb! The other members of Seminar are going to find out her antics at this rate...

She is still on an infinite grid of cells. Each cell has position (i, j) , where the i represents the vertical location and increases as you go down the grid, and the j represents the horizontal location and increases as you go right.

She throws the new bomb at multiple different cells, and she has programmed the bombs to have a *power* of p . Which cells will get caught in the blast of Koyuki's bombs?

Note. When we say *power*, we visually mean this, for $p = 0, 1, 2, 3$:

$p=0$	$p=1$	$p=2$	$p=3$
.....X...
.....X...	..X.X..
.....	...X...	..X.X..	.X....X.
...X...	..X.X..	.X....X.	X.....X
.....	...X...	..X.X..	.X....X.
.....X...	..X.X..
.....X...

[Copy](#)[My submissions](#)**✓ Points:** 100 (partial)**⌚ Time limit:** 15.0s**☰ Memory limit:** 1G**➤ Problem type****▼ Allowed languages**

py3

Here, the bomb is thrown at the cell in the fourth row and the fourth column, and the cells marked with are the cells caught in the blast.

Task Details

Your task is to implement a function named `cells_caught_in_blasts`, which should have the following *signature*:

```
def cells_caught_in_blasts(bs, p):
```

[Copy](#)

The above says that it has two arguments `bs` and `p`.

- `bs` is a tuple of pairs (`tuple`) of length 2) denoting the positions where the bombs are thrown.
- `p` is an integer (`int`).

Each position is a pair `(i, j)` corresponding to (i, j) .

The function must return a `frozenset` of pairs of integers denoting the cells caught in the blast.

Restrictions

- The following symbols can now be used: `min`, `max`, `sum`, `range`, `all`, `any`.
- recursion is *disallowed*.
- comprehensions are allowed.
- at most 6 functions can be defined.
- Your source code must have at most 2000 bytes.

Examples

Example 1 Function Call

```
cells_caught_in_blasts(((0, 0), (1, 1)), 1)
```

[Copy](#)

Example 1 Return Value

```
frozenset((-1, 0), (0, -1), (0, 1), (1, 0), (1, 2), (2, 1))
```

[Copy](#)

Constraints

- The function `cells_caught_in_blasts` will be called at most 8 times.
- $0 \leq p \leq 50,000$
- $|i|, |j| \leq 10^{20}$
- The length of `bs` in one call to `cells_caught_in_blasts` will be at most 8.
- The sum of the lengths of `bs` across all calls to `cells_caught_in_blasts` will be at most 8.

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 25 ❤ points if you solve all test cases where:
 - $p \leq 1$
 - The length of `bs` in one call to `cells_caught_in_blasts` will be at most 2.
- You get 50 🍒 points if you solve all test cases where:
 - $p \leq 60$
- You get 25 ❤ points if you solve all test cases.

Clarifications

[Report an issue](#)

No clarifications have been made at this time.