[CS 11] Prac 7t – Spiral

Problem Statement

Enumerate all lattice points in the plane in the following *spiral* order, starting from the origin:

```
#---#---#---#---#...
|
#   #---#---#---#
|   |           |
#   #   #---#   #
|   |   |   |   |
#   #   #   #   #
|   |       |   |
#   #---#---#   #
|               |
#---#---#---#---#
```

In other words, starting from the origin (marked `0` below):

```
 .  .  .  .  .  .  .
 .  .  9 10 11 12  .
 .  .  8  1  2 13  .
 .  .  7  0  3 14  .
 .  .  6  5  4 15  .
 .  .  .... 17 16  .
 .  .  .  .  .  .  .
```

Submit solution [CS 11] Practice 7 ♥

✓ **Points:** 150 (partial)

Time limit: 6.0s

Memory limit: 1G

Author:
kvatienza (Kevin Atienza)

Problem type

Allowed languages
~~NONE~~, py3

Task Details

Your task is to implement a function called `spiral_coords`. This function takes no parameters.

The function must return a *generator* that generates pairs of `int`s, as described in the problem statement.

Note that your generator must be **as lazy as possible**. It should yield each resulting next element as soon as it has enough information, and it should produce these results while advancing the input generators for as little as possible.

Restrictions

(See 7a 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 900.

Example Calls

Example 1 Function Call

```
[*take(8, spiral_coords())]
```

Example 1 Return Value

```
[(0, 0), (0, 1), (1, 1), (1, 0), (1, -1), (0, -1), (-1, -1),
(-1, 0)]
```

Constraints

When your program is run:

- The function `spiral_coords` will be called at most 200 times.
- At most 500 elements will be consumed from the returned generator.

Scoring

- You get 150 ♥ points if you solve all test cases.

Clarifications

Report an issue

No clarifications have been made at this time.