[CS 11] Prac 7l – 8 Neighbors

Problem Statement

Given the coordinates of a lattice point in the Cartesian plane, output all of its eight neighboring lattice points, starting from the one north of it, and then going clockwise (so the next is northeast, then east etc.).

The neighbors of the point marked `#` are marked `*` below:

```
. . . . . . .
. . . . . . .
. . * * * . .
. . * # * . .
. . * * * . .
. . . . . . .
. . . . . . .
```

Copy

Formally, two points are neighbors in the above sense iff the ∞-norm distance between them is exactly 1.

Task Details

Your task is to implement a function called `neighs8`. This function has two `int` parameters denoting the coordinates of the point, (x, y).

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 400.

Example Calls

Example 1 Function Call

```
[*neighs8(5, 2)]
```

Copy

Example 1 Return Value

```
[(5, 3), (6, 3), (6, 2), (6, 1), (5, 1), (4, 1), (4, 2), (4,
```

Copy

Constraints

When your program is run:

- The function `neighs8` will be called at most 200 times.
- Each input coordinate has absolute value at most 10^{10}.

Scoring

- You get 80 ♥ points if you solve all test cases.

❓ Clarifications

Report an issue

No clarifications have been made at this time.

Submit solution

[CS 11]
Practice 7 ♥

My submissions ♥

✔ **Points:** 80 (partial)

Time limit: 6.0s

Memory limit: 1G

Author:
kvatienza (Kevin Atienza)

Problem type

Allowed languages
~~NONE~~, py3