[CS 11 25.1] HOPE 2f – Know Thy Neighbors

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

Submit solution [CS 11 25.1] HOPE 2

My submissions

Points: 200 (partial)
Time limit: 12.0s
Memory limit: 2G

Problem type

Allowed languages
py3

Problem Statement

As a member of your local hopeowners association, you are tasked to process the residential data in your area.

You are given the following data for some people:

- their name,
- their house number, and
- their house number's coordinates (x, y).

For each **house number**, identify:

- the name of its resident,
- its location, and
- the names of the residents in the house's *neighbors*.

We say that two houses located at (x_1, y_1) and (x_2, y_2) are *neighbors* if $|x_1 - x_2| + |y_1 - y_2| \leq d$.

Task Details

Your task is to implement a function named `process_data`, which should start like this:

```
def process_data(data, d):
```

Here, `data` is a sequence of triples (tuples of length 3) corresponding to the house data entries, and d is as described in the problem statement. Each triple in `data` contains:

- a string denoting a resident's name
- an integer denoting the house number of the house they are in, and
- a pair of integers denoting the house's location.

The function must return a dictionary, where the keys are the house numbers, and the values are dictionaries with three keys:

- `name`, whose value is a string denoting the name of the resident in that house,
- `location`, whose value is a pair of integers denoting the coordinates of the house, and
- `neighbors`, whose value is a list of strings denoting the names of the residents in the neighboring houses (in any order).

Restrictions

- Loops and lists are allowed.
- Sets and dictionaries are allowed.
- Generators and comprehensions are allowed.
- Recursion is allowed.
- Your source code must have at most 800 bytes.

Examples

Example 1 Function Call

```
process_data((
    ("Daw", 210, (20, 30)),
    ("Dawit", 230, (20, 35)),
    ("Dawone", 250, (20, 20)),
), 10)
```

Example 1 Return Value

```
{
    210: {
        "name": "Daw",
        "location": (20, 30),
        "nearby": ["Dawit", "Dawone"],
    },
    230: {
        "name": "Dawit",
        "location": (20, 35),
        "nearby": ["Daw"],
    },
    250: {
        "name": "Dawone",
        "location": (20, 20),
        "nearby": ["Daw"],
    }
}
```

Constraints

Let e be the number of entries you process in each call to `process_data`.

- The function `process_data` will be called at most 200 times.
- $1 \leq e \leq 6{,}000$
- The sum of the es across all test cases is at most $12{,}000$.
- Each name consists of at most 10 uppercase or lowercase English letters.
- Each house number is an integer between 0 and 10^{20}.
- Each coordinate has absolute value at most 10^{20}.
- No name appears in two different entries.

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 80 ♥ $+$ 120 ● points if you solve all test cases.

Clarifications

Report an issue

No clarifications have been made at this time.