[CS 11] Prac 9b – Exam Papers

Problem Statement

Co-teachers Jimmy and Gem have just finished grading the students' exam papers, and the next task is to encode the grades into the class grade sheet. But before that, the papers need to be sorted in alphabetical name order first.

Jimmy and Gem graded different papers, and after they were done, they diligently sorted their papers in alphabetical order themselves. It is now up to you, their student assistant, to combine these papers into one single sorted sequence.

Being the bright student that you are, you decided that you want to automate this process by building a robot that does it for you. You gave it the designation *assistant to the student assistant* (ASA).

Given two sequences of names, one from Jimmy and one from Gem, each guaranteed to be in alphabetical order, please produce the sequence of names in alphabetical order.

Note that name comparison is **case-insensitive**, so `Gem`, `gem`, and `GEM` are all considered "equal". If there are duplicate (case-insensitive) names, you should prioritize taking the ones from Jimmy's sequence before the ones from Gem's.

Task Details

Your task is to implement a function called `merge`. This function has two parameters, both iterables of `str`s:

- the first is the sequence of names from Jimmy.
- the second is the sequence of names from Gem.

The function must return a *generator* that generates `str`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.

If there are duplicate names, **you should prioritize taking the ones from Jimmy's sequence before the ones from Gem's.**

Restrictions

(See 9a for more restrictions)

For this problem in particular:

- Recursion is **disallowed**. (The recursion limit has been greatly reduced.)
- The source code limit is 2000.

Example Calls

Example 1 Function Call

```
[*merge(
    ('Aerith', 'Barret', 'CaitSith', 'Cid', 'Cloud', 'Nanaki',
'Tifa', 'Vincent', 'Yuffie'),
    ('Amarant', 'Eiko', 'Freya', 'Garnet', 'Quina', 'Steiner',
'Vivi', 'Zidane'),
)]
```

Example 1 Return Value

```
[
    'Aerith', 'Amarant', 'Barret', 'CaitSith', 'Cid', 'Cloud',
'Eiko', 'Freya', 'Garnet',
    'Nanaki', 'Quina', 'Steiner', 'Tifa', 'Vincent', 'Vivi',
'Yuffie', 'Zidane',
]
```

Example 2 Function Call

```
[*merge(
    iter(('cid', 'CID', 'cId', 'cid', 'CID', 'cID', 'Cindy')),
    iter(('cid', 'cid', 'CID', 'cid', 'CiD', 'Cidney')),
)]
```

Example 2 Return Value

```
['cid', 'CID', 'cId', 'cid', 'CID', 'cID', 'cid', 'cid', 'CID',
'cid', 'CiD', 'Cidney', 'Cindy']
```

Example 2 Explanation

Note: In case of duplicates such as this example, it is required for you to take the one from the first iterable first.

Constraints

- The function `merge` will be called at most 200 times.
- At most 250 elements will be consumed from the returned generator.
- Each name is a string of at most 8 English letters.
- Each input sequence is sorted.

Scoring

- You get 150 ❤ points if you solve all test cases where:
  - each name is nonempty.
- You get 50 ❤ points if you solve all test cases.

Submit solution [CS 11] Practice 9 ❤

My submissions ❤

Points: 200 (partial)
Time limit: 4.0s
Memory limit: 1G

Author:
kvatienza (Kevin Atienza)

Problem type

Allowed languages
~~NONE~~, py3

Clarifications

Report an issue

No clarifications have been made at this time.