



[CS 11] Prac 7g – Triples

Problem Statement

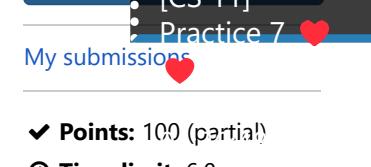
Given a sequence of integers, give all of its *triples* of elements in consecutive locations.

Task Details

Your task is to implement a function called `consec_triples`. This function has a single parameter: an iterable of `int`s.

The function must return a *generator* that generates triples 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.



✓ Points: 100 (partial)

⌚ Time limit: 6.0s

☰ Memory limit: 1G

✎ Author:

kvatienda (Kevin Atienza)

➤ Problem type

▼ Allowed languages

NONE, py3

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 700.

Example Calls

Example 1 Function Call

```
[*consec_triples((3, 1, 4, 1, 5))]
```

Copy

Example 1 Return Value

```
[(3, 1, 4), (1, 4, 1), (4, 1, 5)]
```

Copy

Example 2 Function Call

```
[*consec_triples([3])]
```

Copy

Example 2 Return Value

```
[]
```

Copy

Constraints

When your program is run:

- The function `consec_triples` will be called at most 200 times.
- At most 500 elements will be consumed from the returned generator.
- Each element of the input sequence is a positive integer at most 10^{10} .

Scoring

- You get 100 ❤ points if you solve all test cases.

?

Clarifications

Report an issue

No clarifications have been made at this time.