[CS 11 25.1] Lab 7j – Patas na Pasta

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

Submit solution

[CS 11 25.1] Lab Exercise 7

✓ **Points:** 150 (partial)

Time limit: 4.0s

Memory limit: 1G

› **Problem type**

⌄ **Allowed languages**
py3

Problem Statement

(See 7e for more context)

A bunch of tourists have joined the Pista ng Pasta! Unlike the locals, the tourists want to try out all the pasta equally.

Specifically, a tourist starts at some station i and exits after checking out station j, with $0 \leq i \leq j < n$, and they will get *exactly one* serving from each station, regardless of type, and they will eat it. However, each tourist wants to eat *equally many* servings of each type.

How many distinct pairs of integers (i, j) are there with $0 \leq i \leq j < n$ so that the tourist eats equally many servings of each type?

Task Details

In this problem, you will be provided a module named `pasta` whose contents are exactly as described in D1.

Your task is to implement a function called `count_patas_ranges`. It takes a single argument, a `tuple` or `list` of n `Station`s representing the pasta stations from left to right.

The function must return an `int` denoting the answer.

Restrictions

Note that some names are banned. Here are a few of them: `input`, `type`. This is not an exhaustive list. (If you accidentally use a variable name that turns out to be banned, please rename it.)

The following names are allowed: `map`, `filter`.

The following imports are allowed:

- `count`, `islice`, `chain`, `takewhile`, `starmap` and `zip_longest` from `itertools`.
- `cache`, `lru_cache`, `total_ordering`, `partial`, `reduce` and `wraps` from `functools`.
- `randint`, `randrange` and `choice` from `random`.
- `Fraction` from `fractions`.
- `dataclass` from `dataclasses`.
- `contextmanager` from `contextlib`.
- `Enum`, `auto` from `enum`.

(Read the docs to learn what they do!)

Anonymous functions are allowed.

Inner functions are allowed.

Classes, dataclasses and enums are allowed.

For this problem in particular:

- The following imports are allowed: `Pasta`, `Station` from `pasta`.
- The source code limit is 3000.

Example Calls

Example 1 Function Call

```
count_patas_ranges((
    Station(pasta_type=Pasta.PUTTANESCA, h=32),
    Station(pasta_type=Pasta.PUTTANESCA, h=11),
    Station(pasta_type=Pasta.ANELLETTI, h=2),
    Station(pasta_type=Pasta.ANELLETTI, h=3),
    Station(pasta_type=Pasta.FAGOTTINI, h=5),
    Station(pasta_type=Pasta.PUTTANESCA, h=199),
    Station(pasta_type=Pasta.FAGOTTINI, h=5),
))
```

Copy

Example 1 Return Value

```
2
```

Copy

Testing

To test your program locally, you should create a file called `pasta.py` and save the code above to it. Note that this `pasta.py` is **not** to be submitted! The judge has its own version of `pasta.py`. The `pasta.py` you create is only for your own testing.

Constraints

- The function `count_patas_ranges` will be called at most $60{,}000$ times.
- The sum of ns across all calls will be $\leq 200{,}000$.
- $1 \leq n \leq 200{,}000$
- $0 \leq h_i \leq 10^{20}$

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 50 ❤ points if you solve all test cases where:
  - $n \leq 50$
  - The sum of the ns across all calls will be 500.
- You get 50 ● points if you solve all test cases where:
  - $n \leq 4{,}000$
  - The sum of the ns across all calls will be $8{,}000$.
- You get 50 ● points if you solve all test cases.

Clarifications

Report an issue

No clarifications have been made at this time.