[CS 11 25.1] Lab 7c – Man in the Middle

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

Problem Statement

Cryptographers and security experts are very wary of different types of attacks in a cryptosystem. The most dangerous of them all is the man-in-the-middle attack. In this attack, several men arrange themselves by height. And then the man in the middle attacks.

For their safety, cryptographers follow a piece of advice from Michael Jackson that they misheard. Whenever they need to defend their cryptosystem, they're starting with the man in the middle. In particular, they want to determine which one among a group of men will become the man in the middle. Furthermore, they want to determine this as each man enters the group. Note that the man in the middle may change depending on the height of the man that enters.

Given the height of each man that enters, determine the height of the man in the middle when they arrange themselves by height. If there are an even number of men, then there's no man in the middle to attack.

Task Details

Your task is to implement a function called `make_group`. It takes no arguments and initialized an *empty* group.

The function must return a function. Calling this function represents a new man entering the group. It takes in a single argument, an `int` h denoting the height of the man.

It must return an `int` denoting the height of the man in the middle, or `None` if there is no man in the middle (because there are an even number of men in the group).

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 source code limit is 3000.

Example Testing

Note: This assumes that your submission has filename `hope3e1.py`. Write this testing code in a separate file, say `test_hope3e1.py`, and run it to test your code.

Example 1 Testing

Copy

```
from hope3e1 import make_group

def verify(condition: bool, message: str = "verification
failed"):
    assert condition, message

enter_group1 = make_group()
enter_group2 = make_group()

verify(enter_group1(3) == 3)
verify(enter_group2(3) == 3)
verify(enter_group2(1) is None)
verify(enter_group1(1) is None)
verify(enter_group2(6) == 3)
verify(enter_group1(4) == 3)
verify(enter_group1(1) is None)
verify(enter_group2(2) is None)
verify(enter_group2(2) == 2)
verify(enter_group1(5) == 3)
verify(enter_group1(9) is None)
verify(enter_group2(7) is None)
verify(enter_group1(2) == 3)
verify(enter_group2(7) == 3)
verify(enter_group2(6) is None)
verify(enter_group1(6) is None)
```

Constraints

- The function `make_group` will be called at most $60{,}000$ times.
- The total size of all groups at the end will be $\leq 300{,}000$.
- $1 \leq h \leq 10^8$

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 60 ❤️ points if you solve all test cases where:
  - The total size of all groups at the end will be 50.
- You get 100 🔴 points if you solve all test cases where:
  - The total size of all groups at the end will be $8{,}000$.
- You get 50 🔴 points if you solve all test cases where:
  - Each person that comes in is either the tallest or the shortest man so far in the group.
- You get 50 💔 points if you solve all test cases.

For the first category above, you only get 10 points if you solve the smoke tests but not all the tests in that category.

Clarifications

Report an issue

No clarifications have been made at this time.