[CS 11] Prac 1f – Rice Price Rise

oj.dcs.upd.edu.ph/problem/cs11prac1f

Problem Statement

There's a wide variety of rice variants on sale at Wall Market—dinorado, sinandomeng, angelica, kokuyo, to name a few.

However, you don't care about any of that. You actually don't even know the differences between these rice variants. "Rice is just rice," you say. "They all taste the same."

Thus, you simply want to buy the cheapest rice available.

There are several sacks of rice in the market in a sequence. Each of these sacks has a price, in pesos. You want to buy the cheapest sack of rice. If there are multiple sacks with the same cheapest price, you want to buy the leftmost one among them.

Once you buy that sack of rice, it will be removed from the sequence.

What is the price of the sack of rice you will buy, and what would the sequence look like when that sack of rice is removed?

Task Details

Your task is to implement a function called `buy_cheapest`. This function has a single parameter, a `tuple` of `int`s representing the prices of the sacks of rice in sequence.

The function must return a pair (that is, a 2-tuple).

- The first element of the pair is an `int` denoting the price of the sack of rice you will buy.
- The second element of the pair is a `tuple` of `int`s representing what the sequence looks like when that sack of rice is removed.

Restrictions

For this problem:

- Assignment is allowed.
- Recursion is allowed.
- Up to 66 function definitions are allowed.

- Comprehensions are **disallowed**.
- `range` is **disallowed**.
- The `abs` symbol is now allowed.
- The source code limit is 10001000.

Example Calls

Example 1 Function Call

Copy

```
buy_cheapest((31, 41, 59, 26, 53))
```

Example 1 Return Value

Copy

```
(26, (31, 41, 59, 53))
```

Example 2 Function Call

Copy

```
buy_cheapest((33, 11, 44, 11, 55))
```

Example 2 Return Value

Copy

```
(11, (33, 44, 11, 55))
```

Constraints

- The function `buy_cheapest` will be called at most 1,0001,000 times.
- The argument to each function call will have at most 2020 elements, and each element of it is a positive integer at most 100100.

Scoring

- You get 100100 ❤️ points if you solve all test cases where:
  - All prices are distinct.
- You get 5050 ❤️ points if you solve all test cases.

Report an issue

Clarifications

No clarifications have been made at this time.