[CS 11 25.1] Lab 7b – Window Shopping

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

Problem Statement

You want to buy a bunch of items from the Poké Mart. There are n items lined up in a row, with prices $p_1, p_2, \cdots, p_n$.

Define a function that will answer q questions of the following form:

You want to buy k consecutive items among the n items. The amount you end up paying is the total price of the k items you select. Among the $n - k + 1$ possible ways to do this, how much do you have to pay for the i^{th} most expensive way?

Note that k is **fixed** across questions; only i varies.

Task Details

Your task is to implement a function named `window_shopping`. It should have two arguments p and k.

- p is a sequence of integers (`int`s) denoting the prices of the n items.
- k is an integer the number of consecutive items you want to buy.

The function must return a **function** that takes in an integer i and returns an integer denoting the amount you have to pay for the i^{th} most expensive way to select k consecutive items.

It is guaranteed that when this function is called, $1 \leq i \leq n - k + 1$.

Name your file `lab06b.py` and your testing file `test_lab06b.py`.

Restrictions

- The following symbols are now allowed: `map`, `filter`
- The following imports are now 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`.
- Anonymous functions are now allowed.
- Inner functions are allowed.
- Classes, dataclasses, and enums are allowed.
- Recursion is allowed.
- Loops are allowed.
- Generators and comprehensions are allowed.
- Your source code must have at most 700 bytes.

Example Testing

Here's an example testing file.

```
# pyright: strict

from lab06b import window_shopping

ith_most_expensive = window_shopping((1, 2, 3, 4), 2)
assert ith_most_expensive(1) == 7

# TODO add more tests here
```

Copy

Constraints

- The function `window_shopping` will be called at most $70{,}000$ times.
- $1 \leq k \leq n \leq 350{,}000$
- $1 \leq q \leq 350{,}000$
- The sum of n across all calls to `window_shopping` will be $\leq 350{,}000$.
- The sum of q across all calls to `window_shopping` will be $\leq 350{,}000$.
- $1 \leq p_i \leq 10^{20}$

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:
  - $n \leq 50$
  - $q \leq 50$
  - The sum of n across all calls to `window_shopping` will be ≤ 800.
  - The sum of q across all calls to `window_shopping` will be ≤ 800.
- You get 25 ❤️ points if you solve all test cases where:
  - $n \leq 400$
  - The sum of n across all calls to `window_shopping` will be ≤ 800.
  - The sum of q across all calls to `window_shopping` will be ≤ 800.
- You get 120 🔴 points if you solve all test cases where:
  - $n \leq 6{,}000$
  - $q \leq 6{,}000$
  - The sum of n across all calls to `window_shopping` will be $\leq 12{,}000$.
  - The sum of q across all calls to `window_shopping` will be $\leq 12{,}000$.
- You get 60 🔴 points if you solve all test cases.

Sub[illegible] solution

[CS 11 25.1] Lab Exercise 7

My submissions

Points: 255 (partial)
Time limit: 6.0s
Memory limit: 2G

Problem type

Allowed languages
py3

Clarifications

Report an issue

No clarifications have been made at this time.