

[CS 11 25.1] Lab 5c – Arrays and Products

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

Problem Statement

You are given an array of n positive integers $[a_1, a_2, \dots, a_n]$.

Compute the number of *subarrays* that have a product less than or equal to p .

A *subarray* is a nonempty sequence of consecutive elements of the array.

Task Details

Your task is to implement a function named `num_subarrays`, which should have the following *signature*:

```
def num_subarrays(a, p):
```

Copy

The above says that it has two arguments a and p .

- a is a sequence of n integers (`int`s).
- p is an integer.

The function must return an integer denoting the number of subarrays of a with product $\leq p$.

Restrictions

- The following symbols can be used:
 - `list`, `set`, `dict`, `enumerate`, `append`, `pop`, `extend`, `remove`, `sort`, `sorted`, `insert`, `clear`, `reverse`, `reversed`, `iter`, `next`, `zip`.
- The following imports are allowed:
 - `count` and `islice` from `itertools`.
- Loops are allowed.
- Generators and comprehensions are allowed.
- Recursion is *disallowed*.
- Your source code must have at most 700 bytes.

Examples

Example 1 Function Call

```
num_subarrays((2, 1, 2), 3)
```

Copy

Example 1 Return Value

```
5
```








Copy

Constraints

- The function `num_subarrays` will be called at most 70,000 times.
- $0 \leq n \leq 350,000$
- The sum of n across all calls to `num_subarrays` will be $\leq 350,000$.
- $1 \leq a_i \leq 10^{10}$
- $1 \leq p \leq 10^{10}$

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 15  points if you solve all test cases where:
 - $p = 1$
 - $n \leq 400$
 - The sum of n across all calls to `num_subarrays` will be ≤ 800 .
- You get 20  points if you solve all test cases where:
 - $n \leq 400$
 - The sum of n across all calls to `num_subarrays` will be ≤ 800 .
- You get 20  points if you solve all test cases where:
 - $p = 1$
 - $n \leq 6,000$
 - The sum of n across all calls to `num_subarrays` will be $\leq 6,000$.
- You get 25  points if you solve all test cases where:
 - $n \leq 6,000$
 - The sum of n across all calls to `num_subarrays` will be $\leq 6,000$.
- You get 25  points if you solve all test cases where:
 - $p = 1$
- You get 50  points if you solve all test cases where:
 - $p \leq 3$
- You get 60  points if you solve all test cases.

Clarifications

No clarifications have been made at this time.


Report an issue


Submit solution


[CS 11 25.1]



Lab Exercise 5

[My submissions](#)

 **Points:** 215 (partial)

 **Time limit:** 7.0s

 **Memory limit:** 2G

-  **Problem type**
-  **Allowed languages**
py3