



[CS 11 25.1] Mock HOPE 2b – Course Cutoff

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

Problem Statement

BS Computer Science is quite popular these days! Unfortunately, the infamous Department of Computer Science in UP Diliman can only accept up to c applicants.

Naturally, they will accept the c applicants with the highest UPCAT scores. The UPCAT score is a positive integer. For example, this can be done by sorting the applicants by UPCAT score, and taking the last c applicants (or less, if there are fewer than c applicants). The **cutoff** score is then defined as the lowest score among the accepted applicants.

There are n applicants who applied in order, one by one. We number them 0 to $n - 1$. For each applicant, what would be the cutoff score if the cutoff is computed right after that applicant submits their application?

Task Details

Your task is to implement a function called `partial_cutoffs`. It has a two arguments:

- a `tuple` of n `int`s denoting the input sequence.
- the `int` c .

It must return a `list` of n `int`s denoting the answers. Specifically, for each i from 0 to $n - 1$, the element at index i is the cutoff score if the application if the cutoff were to be computed right after applicant i submits their application.

Restrictions

Note that some names are banned.

For this problem:

- Loops and lists are allowed.
- Sets and dictionaries are allowed.
- Generators and comprehensions are allowed.
- Recursion is **disallowed**. (The recursion limit has been greatly reduced.)
- Up to 16 function definitions are allowed.
- The source code limit is 2000.

Example Calls

Example 1 Function Call

```
partial_cutoffs((3, 1, 4, 1, 5, 9, 2), 3)
```

Copy

Example 1 Return Value

```
[3, 1, 1, 1, 3, 4, 4]
```

Copy

Constraints

- The function `partial_cutoffs` will be called at most 60,000 times.
- The sum of ns across all calls will be $\leq 250,000$.
- $1 \leq c \leq n \leq 250,000$
- The UPCAT score is an integer between 1 and 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 40 ❤ points if you solve all test cases where:
 - $n \leq 60$
 - The sum of the ns across all calls will be 400.
- You get 50 💔 points if you solve all test cases where:
 - $n \leq 400$
 - The sum of the ns across all calls will be 400.
- You get 60 💔 points if you solve all test cases where:
 - $n \leq 4,000$
 - The sum of the ns across all calls will be 4,000.
- You get 50 💔 points if you solve all test cases.

Clarifications

Report an issue

No clarifications have been made at this time.