beCP
2022
Task 1.2: Rangemin (rangemin)

Author: Damien Galant
Preparation: Damien Galant, Victor Dheur, Pierre Vandenhove
Time limit: 1 s Memory limit: 512 MB

Note: This task is interactive. Please look at the special instructions to implement, compile and test your program. Don't hesitate to ask the staff for help.

There are n distinct positive integers $a_0, a_1, \cdots, a_{n-1}$. Your goal is to find *the index* of the minimum among those integers. However, you do not have directly access to those integers, but only to the function `range_min(i)` (see below). You want to call this function *as few times as possible*.

Functions to call

The function `range_min(i)` returns the minimum *value* among the integers $a_i, a_{i+1}, \cdots, a_{i+k-1}$. Once you have found the *index* of the minimum among $a_0, a_1, \cdots, a_{n-1}$, you should call the function `found_min(i)` once with the index of the minimum.

Functions to implement

You should implement the function `find_min(n, k)`. This function is given the values of n of k. You should make calls to the `range_min` function to find the index of the minimum among $a_0, a_1, \cdots, a_{n-1}$. Once it is found, you should call `found_min` once with the value of this index then stop your program.

Remark: not calling the `range_min` at all on an input will result in a "wrong answer'" verdict, even if you find (by pure luck) the correct index of the minimum.

General limits

All values $a_0, a_1, \cdots, a_{n-1}$ are *distinct* integers such that $1 \leq a_i \leq 100\,000$ for every $0 \leq i < n$.

You can call the `range_min` function at most 100 000 times (this is a practical limitation, a solution calling `range_min` this many times would have a very small score anyway, see section "Scoring").

Additional constraints

Subtask	Points	Constraints
A	20	$n = 100$, $k = 1$
B	20	$n = 100$, $k = 2$
C	60	$n = 1000$, $1 \leq k \leq 500$

Example 1

Here is a possible sequence of events assuming $n = 5$, $k = 2$, $a_0 = 8$, $a_1 = 10$, $a_2 = 4$, $a_3 = 5$ and $a_4 = 9$.

Your call	Received Answer
`range_min(0)`	$\min(8, 10) = 8$
`range_min(3)`	$\min(5, 9) = 5$
`range_min(2)`	$\min(4, 5) = 4$
`found_min(2)`	Index found!

Remark: note that this small example is not included in any subtask (among A, B and C).

Scoring

The scoring for this tasks works as follows:

- The index of the minimum must be found for every testcase of a subtask in order to get points for this subtask.
- When this is the case, let us denote by S the number of calls to `range_min` performed by your solution and by B the number of calls to `range_min` performed by the solution prepared by the beCP judges. The *ratio* for this testcase is defined as $\min(B/S, 1)$. On CMS, your ratio will be shown for each testcase where the index of the minimum was correctly found.
- Finally, your score for a subtask is given by the total number of points for this subtask, multiplied by $0.2 + 0.8 * r$, where r is the *lowest* ratio among testcases for this subtask.

For instance, if the lowest ratio among testcases of subtask B is equal to 0.4, then your score for subtask B will be equal to $20 * (0.2 + 0.8 * 0.4) = 10.4$.

Compilation information

This is an interactive task. Follow the following steps to compile and test manually your program. Ask help to a responsible if necessary.

- Open a terminal
- To display the list of files present in a folder, use command `ls`;
- To change directories to the skeleton of this task, use the command `cd repertory_name`;
- To compile and execute programs written in C++, the commands are as follows:
  - Compilation:
    ```
    g++ -std=c++11 -Wall -Werror -Wshadow grader.cpp rangemin.cpp
    ```
  - Execution: `./a.out < input.in`

The result of your program will be printed to the console: "FOUND THE INDEX AFTER X CALLS" if the index of the minimum was found and "WRONG INDEX WAS GIVEN" if the index you provide is wrong. Other messages may be printed if the arguments you give to `range_min` or `found_min` are wrong. If you want to modify the file `input.in`, know that the first line contains n and k and the second line contains the n integers $a_0, a_1, \cdots, a_n$. Make sure that those integers are distinct and between 1 and 100 000!

Remarks

- You only need to submit the file `rangemin.cpp`. It must implement function `find_min` described above.
- **Do not print anything to the console in the program you submit (`printf`, `cout`).** You can use `cerr` in C++ to display debug information.