[CS 11] Prac 11e – Queuery

Problem Statement

You need to maintain a *queue* of integers. The queue starts with exactly n elements from front to back.

Whenever a new integer is enqueued at the back, you must find the minimum of the elements of the queue, and then after that, dequeue the element in front.

You must process up to q operations of the previous kind.

Submit solution

[CS 11] Practice 11 ♥

Points: 300 (partial)
Time limit: 8.0s
Memory limit: 1G

Author: kvatienza (Kevin Atienza)

Problem type

Allowed languages
~~NONE~~, py3

Task Details

Your task is to implement a function called `rolling_queue`. The function takes a single argument, a `tuple` or `list` of n integers denoting the initial values, from front to back.

The function must return a function that takes in a single `int` argument. When this function is called, the given `int` must be enqueued, and then the frontmost element must be dequeued. The function must also return the minimum element of the queue right before the dequeue operation.

Restrictions

(See 11a for more restrictions)

For this problem in particular:

- The source code limit is 3000.

Example Testing

Note: This assumes that your submission has filename `prac11e.py`. Write this testing code in a separate file, say `test_prac11e.py`, and run it to test your code.

Example 1 Testing

```
from prac11e import rolling_queue

def verify(condition: bool, message: str = "verification
failed"):
    assert condition, message

op1 = rolling_queue([3, 1, 6])
op2 = rolling_queue([3, 1])

verify(op1(2) == 1)
verify(op2(4) == 1)
verify(op2(1) == 1)
verify(op1(2) == 1)
verify(op2(5) == 1)
verify(op1(7) == 2)
verify(op1(7) == 2)
verify(op2(9) == 1)
verify(op2(2) == 2)
verify(op1(6) == 2)
verify(op1(6) == 6)
verify(op2(6) == 2)
verify(op1(0) == 0)
verify(op2(5) == 2)
verify(op2(3) == 3)
verify(op1(1) == 0)
verify(op2(5) == 3)
```

Copy

Constraints

- The function `rolling_queue` will be called at most $70{,}000$ times.
- $0 \le n \le 350{,}000$
- $0 \le q$
- The sum of the ns will be at most $350{,}000$.
- The sum of the qs will be at most $350{,}000$.
- Each integer will have absolute value at most 10^{10}.

Scoring

- You get 150 ♥ points if you solve all test cases where:
  - $n \le 50$
  - the sum of the ns is at most 600.
  - the sum of the qs is at most 600.
- You get 50 ♥ points if you solve all test cases where:
  - $n \le 5{,}000$
  - the sum of the ns is at most $5{,}000$.
  - the sum of the qs is at most $5{,}000$.
- You get 100 💔 points if you solve all test cases.

Clarifications

Report an issue

No clarifications have been made at this time.