

Stack Sort

Julie works as an engineer at a factory and while testing, is in charge of sorting blocks of different sizes into the correct order from smallest to largest.

Each set of blocks is composed of n blocks, of sizes 1, 2, ..., n but in a random order.

Due to circumstances unexplained to her, she is required to sort these blocks in a very specific way.

Blocks come in on an input queue and must be placed on a LIFO stack and then a FIFO output queue in order to be sorted with a weird set of rules:

At any one time a Julie can either:

- Pull the next block off the head of the input queue and place it on the top of the sorting stack.
- Take a block off of the top of sorting stack and place it into an output queue with FIFO ordering.

Once all of the blocks have entered the output queue, this round is over.

Julie then checks to see if the blocks are sorted, if not, Julie repeats these stacking and placing operations for another round. Julie continues doing this until the blocks are sorted.

Julie wonders, for some random assortment of blocks of sizes 1, 2, ..., n ; what is the number of rounds needed to sort using this method?

Input:

First a line with an integer T , the number of test cases to follow.

T test cases follow. Each test case starts on a new line and has on the first line an integer n , the number of blocks for this test case.

On the following line there are n integers, a_1, a_2, \dots, a_n the size of each block in the input queue where a_1 is the head of the queue.

Output:

Output the minimum number of "rounds" needed to sort the blocks in ascending or using by following the rules.

Note: If a set of blocks comes in on a queue already sorted, 0 rounds are needed to sort them.

Constraints on actual problem:

$T = 250$

$3 \leq n \leq 100$

Sample Input:

6

7

3 7 6 5 4 2 1

8

3 7 4 8 1 2 5 6

2

2 1

7

4 6 5 2 3 1 7

10

2 4 3 9 1 10 5 7 8 6

5

1 2 3 4 5

Sample Output:

2

4

1

2

4

0