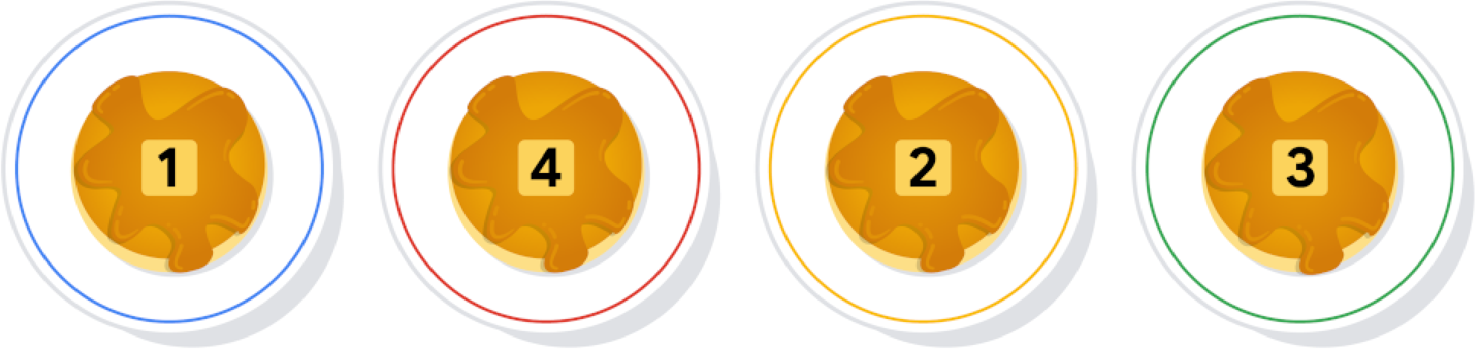
Google Code Jam '22 Round 1B Problem A - Pancake Deque

Time limit: 20.0s **Memory limit:** 1G

Pancakes are normally served in stacks, but the Infinite House of Pancakes embraces change! The restaurant's new advertising hook is to serve the pancakes from a deque, or double-ended queue.

You are a server at the restaurant, and your job is to serve every pancake in the deque. Customers will arrive one at a time, and each one gets a single pancake. You must serve each customer either the leftmost or rightmost pancake in the deque; the choice is yours. When a pancake is served, it disappears from the deque, exposing the pancake that was next to it. Or, once there is only one pancake left, your only choice is to serve that one, and then your job is complete!



Each pancake has a deliciousness level. Because customers do not get to choose which pancakes they get, each customer only has to pay for their pancake if it is at least as delicious as *each* of the pancakes that *all* of the previous customers got. (The first customer always pays for their pancake since in that case there are no previous customers.)

How many customers will pay for their pancake, if you serve the pancakes in an order that maximizes that number?

Input Specification

The first line of the input gives the number of test cases, T. T test cases follow. Each test case is described with two lines. The first line of a test case contains a single integer N, the number of pancakes in the pancake deque. The second line of a test case contains N integers $D_1, D_2, \ldots, D_N$, where D_i is the deliciousness level of the i^{th} pancake from the left in the deque.

Output Specification

For each test case, output one line containing `Case #x: y`, where x is the test case number (starting from 1) and y is the number of customers who pay for their pancakes, if you serve the pancakes in an order that maximizes that number.

Limits

Time limit: 20 seconds.

Memory limit: 1 GB.

$1 \leq T \leq 100$.

$1 \leq D_i \leq 10^6$, for all i.

Test Set 1

$2 \leq N \leq 20$.

Test Set 2

$2 \leq N \leq 100$.

Test Set 3

$2 \leq N \leq 10^5$.

Sample Input

```
4
2
1 5
4
1 4 2 3
5
10 10 10 10 10
4
7 1 3 1000000
```

Sample Output

```
Case #1: 2
Case #2: 3
Case #3: 5
Case #4: 2
```

In Sample Case #1, there are two possible orders in which you can serve the pancakes. If you serve the pancake with deliciousness level 5 first, only that one is paid for. If you serve the pancake with deliciousness level 1 first, both are paid for.

Sample Case #2 is the image shown in the problem statement. The following are the possible orders (by deliciousness level) in which the pancakes can be served. The underlined pancakes are the ones that customers pay for.

- $\underline{1}, \underline{4}, 2, 3$
- $\underline{1}, \underline{4}, 3, 2$
- $\underline{1}, \underline{3}, \underline{4}, 2$

- $\underline{1}, \underline{3}, 2, \underline{4}$
- $\underline{3}, 1, \underline{4}, 2$
- $\underline{3}, 1, 2, \underline{4}$
- $\underline{3}, 2, 1, \underline{4}$
- $\underline{3}, 2, \underline{4}, 1$

As you can see, there are some orders in which 3 pancakes are paid for, and none in which all 4 are.

In Sample Case #3, all pancakes are paid for regardless of the serving order.

In Sample Case #4, regardless of which pancake you serve first, the two in the middle will never be paid for. The best you can do is serve the pancake with deliciousness 7 before the pancake with deliciousness $1\,000\,000$.