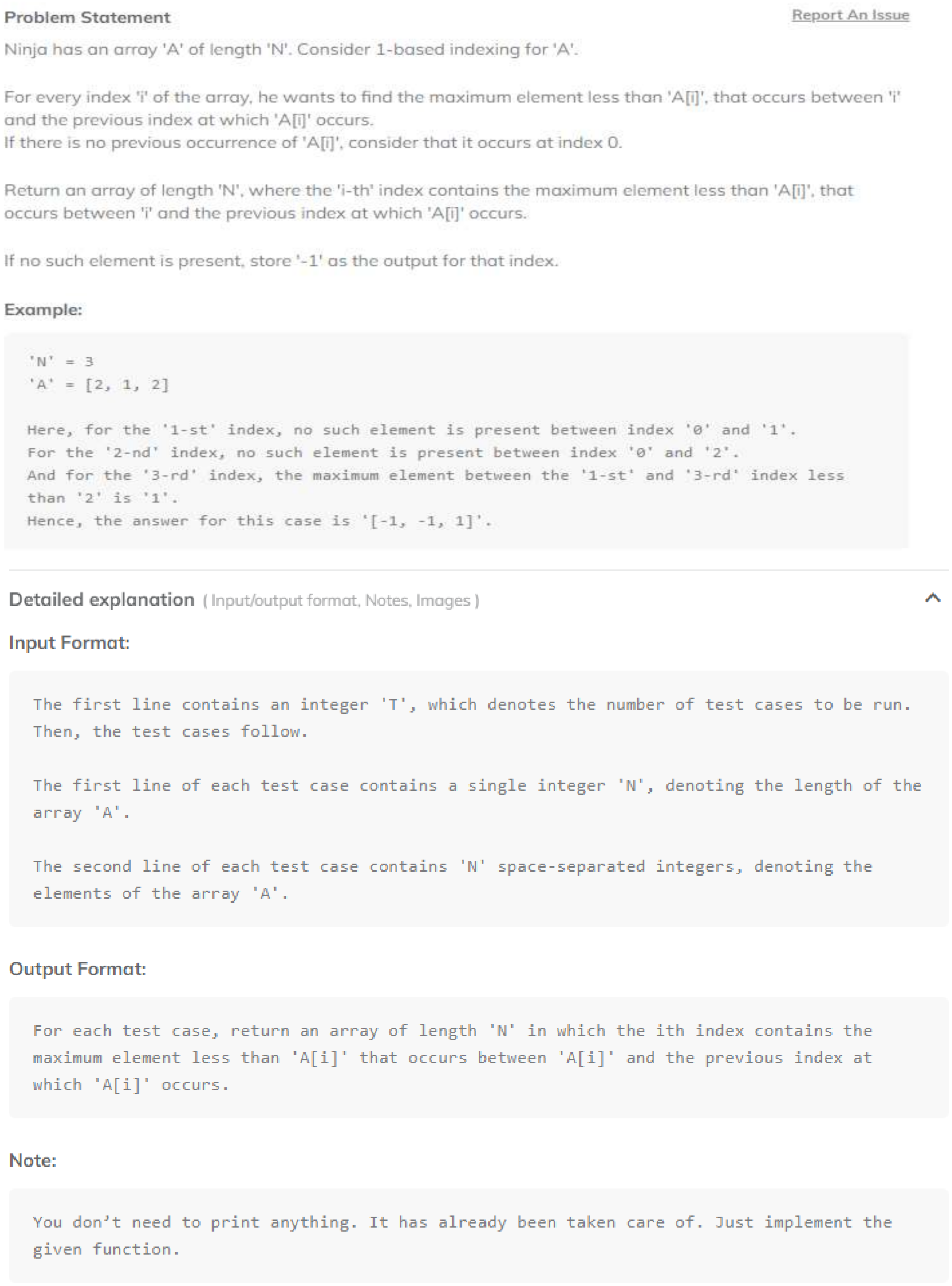
Less-Max Element



Problem Statement

Report An Issue

Ninja has an array 'A' of length 'N'. Consider 1-based indexing for 'A'.

For every index 'i' of the array, he wants to find the maximum element less than 'A[i]', that occurs between 'i' and the previous index at which 'A[i]' occurs.
If there is no previous occurrence of 'A[i]', consider that it occurs at index 0.

Return an array of length 'N', where the 'i-th' index contains the maximum element less than 'A[i]', that occurs between 'i' and the previous index at which 'A[i]' occurs.

If no such element is present, store '-1' as the output for that index.

Example:

```
'N' = 3
'A' = [2, 1, 2]

Here, for the '1-st' index, no such element is present between index '0' and '1'.
For the '2-nd' index, no such element is present between index '0' and '2'.
And for the '3-rd' index, the maximum element between the '1-st' and '3-rd' index less
than '2' is '1'.
Hence, the answer for this case is '[-1, -1, 1]'.
```

Detailed explanation (Input/output format, Notes, Images)

Input Format:

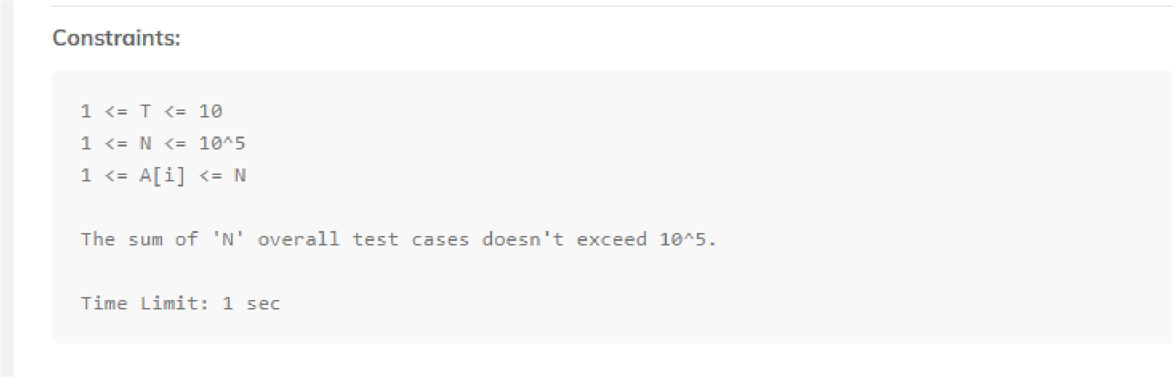
```
The first line contains an integer 'T', which denotes the number of test cases to be run.
Then, the test cases follow.

The first line of each test case contains a single integer 'N', denoting the length of the
array 'A'.

The second line of each test case contains 'N' space-separated integers, denoting the
elements of the array 'A'.
```

Output Format:

```
For each test case, return an array of length 'N' in which the ith index contains the
maximum element less than 'A[i]' that occurs between 'A[i]' and the previous index at
which 'A[i]' occurs.
```

Note:

```
You don't need to print anything. It has already been taken care of. Just implement the
given function.
```



Constraints:

```
1 <= T <= 10
1 <= N <= 10^5
1 <= A[i] <= N

The sum of 'N' overall test cases doesn't exceed 10^5.

Time Limit: 1 sec
```

Sample Input 1:

```
2
2
1 2
3
4 6 7
```

Sample Output 1:

```
-1 1
-1 4 6
```

Explanation Of Sample Input 1:

For test case 1:
Here, for the '1-st' index, no such element is present between index '0' and '1'.
And for the '2-nd' index, the maximum element between index '0' and '2' less than '2' is '1'.
Hence, the answer for this case is [-1, 1].

For test case 2:
Here, for the '1-st' index, no such element is present between index '0' and '1'.
For the '2-nd' index, the maximum element between index '0' and '2' less than '6' is '4'.
And for the '3-rd' index, the maximum element between the '0-th' and the '3-rd' index less than '7' is '6'.
Hence, the answer for this case is [-1, 4, 6].

Sample Input 2:

```
2
6
3 2 1 2 4 3
4
3 4 1 2
```

Sample Output 2:

```
-1 -1 -1 1 3 2
-1 3 -1 1
```