### Next Greater Element

Given an array **arr[ ]** of size **N** having elements, the task is to find the next greater element for each element of the array in order of their appearance in the array.  
Next greater element of an element in the array is the nearest element on the right which is greater than the current element.  
If there does not exist next greater of current element, then next greater element for current element is -1. For example, next greater of the last element is always -1.

**Example 1:**

**Input**:

N = 4, arr[] = [1 3 2 4]

**Output**:

3 4 4 -1

**Explanation**:

In the array, the next larger element

to 1 is 3 , 3 is 4 , 2 is 4 and for 4 ?

since it doesn't exist, it is -1.

**Example 2:**

**Input**:

N = 5, arr[] [6 8 0 1 3]

**Output**:

8 -1 1 3 -1

**Explanation**:

In the array, the next larger element to

6 is 8, for 8 there is no larger elements

hence it is -1, for 0 it is 1 , for 1 it

is 3 and then for 3 there is no larger

element on right and hence -1.

Java code

import java.util.\*;

import java.lang.\*;

import java.io.\*;

class CodingMaxima {

public static void main (String[] args) throws IOException {

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int t = Integer.parseInt(br.readLine().trim());

while(t-->0){

int n = Integer.parseInt(br.readLine().trim());

String inputLine[] = br.readLine().trim().split(" ");

long[] arr = new long[n];

for(int i=0; i<n; i++)arr[i]=Long.parseLong(inputLine[i]);

long[] res = new Solution().nextLargerElement(arr, n);

for (int i = 0; i < n; i++)

System.out.print(res[i] + " ");

System.out.println();

}

}

}

class Solution

{

//Function to find the next greater element for each element of the array.

public static long[] nextLargerElement(long[] arr, int n)

{

long[] result = new long[n];

Arrays.fill(result, -1);

Stack<Integer> stack = new Stack<>();

for (int i = 0; i < n; i++) {

while (!stack.isEmpty() && arr[i] > arr[stack.peek()]) {

result[stack.pop()] = arr[i];

}

stack.push(i);

}

return result;

}

}