503. Next Greater Element II

Given a circular integer array `nums` (i.e., the next element of `nums[nums.length - 1]` is `nums[0]`), return *the **next greater number** for every element in* `nums`.

The **next greater number** of a number `x` is the first greater number to its traversing-order next in the array, which means you could search circularly to find its next greater number. If it doesn't exist, return `-1` for this number.

Example 1:

```
Input: nums = [1,2,1]
Output: [2,-1,2]
Explanation: The first 1's next greater number is 2; 
The number 2 can't find next greater number. 
The second 1's next greater number needs to search circularly, which is also 2.
```

Example 2:

```
Input: nums = [1,2,3,4,3]
Output: [2,3,4,-1,4]
```

Constraints:

- `1 <= nums.length <= 10<sup>4</sup>`
- `-10<sup>9</sup> <= nums[i] <= 10<sup>9</sup>`

```
class Solution:
    def nextGreaterElements(self, arr: List[int]) -> List[int]:
        stack = []


        for i in range(len(arr)-2,-1,-1):
            while len(stack) > 0 and stack[-1] <= arr[i]:
                stack.pop()
            # ansLeft[i] = stack[-1] if len(stack) else -1
            stack.append(arr[i])


        ansRight = [None] * len(arr)
        for i in range(len(arr)-1,-1,-1):
            while len(stack) > 0 and stack[-1] <= arr[i]:
                stack.pop()
            ansRight[i] = stack[-1] if len(stack) else -1
            stack.append(arr[i])


        return ansRight
```