

单调栈：

题目描述：

503. 下一个更大元素 II

难度 中等

477



给定一个循环数组（最后一个元素的下一个元素是数组的第一个元素），输出每个元素的下一个更大元素。数字 x 的下一个更大的元素是按数组遍历顺序，这个数字之后的第一个比它更大的数，这意味着你应该循环地搜索它的下一个更大的数。如果不存在，则输出 -1 。

示例 1:

输入: $[1, 2, 1]$

输出: $[2, -1, 2]$

解释: 第一个 1 的下一个更大的数是 2 ；

数字 2 找不到下一个更大的数；

第二个 1 的下一个最大的数需要循环搜索，结果也是 2 。

注意: 输入数组的长度不会超过 10000 。

题目分析：

1. 对于「找最近一个比当前值大/小」的问题，都可以使用单调栈来解决。

- 相等情况的处理
- 循环的处理：偏暴力的思想

代码分析：

```
class Solution {
public:
    vector<int> nextGreaterElements(vector<int>& nums) {
        int n=nums.size();
        vector<int>res(n,-1);
        stack<int>stk;
        //最大的数不用管，已经设置为-1了
        //暴力思想：最多遍历两遍
        for(int i=0;i<2*n-1;i++){
            while(!stk.empty()&&nums[stk.top()]<nums[i%n]){
                res[stk.top()]=nums[i%n];
                stk.pop();
            }
            stk.push(i%n);
        }
        return res;
    }
};
```