

{% note info %} **摘要** Title: 830. 单调栈 Tag: 单调栈 Memory Limit: 64 MB Time Limit: 1000 ms {% endnote %}

Powered by: NEFU AB-IN

[Link](#)

@TOC

## 830. 单调栈

### • 题意

给定一个长度为  $N$  的整数数列，输出每个数左边第一个比它小的数，如果不存在则输出  $-1$ 。

### • 思路

**单调栈**板子题，复杂度 $O(n)$

**单调栈：给定一个序列，求每一个数的左边离他最近的（最小/最大）数是什么**

{% note info %} **每一个数的右边离他最近的（最小/最大）数？** 倒着遍历，不用改变**符号** {% endnote %}

思路就是当某个元素 $a$ 被遍历时，先在栈中把比 $a$ 大的数弹出去，则此时如果栈中还有元素，那么一定是比 $a$ 小的

{% note info %} **为什么大的要弹出去？** 因为最后此元素 $a$ 一定会加进去，而 $a$ 的下一个元素如果选择比它小的左边的元素，肯定选择 $a$ 是要比选择比 $a$ 大的要优 {% endnote %}

注意：

- 遍历数组时用不到下标
- 别忘了最后将元素加入栈
- **栈用列表模拟即可**
  - 栈顶为 `stk[-1]`
  - 入栈和出栈分别为 `append` 和 `pop`

### • 代码

```
'''
Author: NEFU AB-IN
Date: 2022-02-26 20:13:53
FilePath: \ACM\Acwing\830.py
LastEditTime: 2022-02-26 22:36:21
'''

if __name__ == "__main__":
    n = int(input())
    nums = list(map(int, input().split()))
    stk = []
```

```
for num in nums:
    while stk and stk[-1] >= num: #改成 <= 就是求每个数左边第一个比它 大 的
        stk.pop()
    print(stk[-1] if stk else -1, end=" ")
    stk.append(num)
```