

## 第38题——下一个更大元素

给你两个 没有重复元素 的数组 `nums1` 和 `nums2`，其中`nums1` 是 `nums2` 的子集。

请你找出 `nums1` 中每个元素在 `nums2` 中的下一个比其大的值。

`nums1` 中数字 `x` 的下一个更大元素是指 `x` 在 `nums2` 中对应位置的右边的第一个比 `x` 大的元素。如果不存在，对应位置输出 `-1`。

示例 1:

输入: `nums1 = [4,1,2]`, `nums2 = [1,3,4,2]`。

输出: `[-1,3,-1]`

解释:

对于 `num1` 中的数字 `4`，你无法在第二个数组中找到下一个更大的数字，因此输出 `-1`。

对于 `num1` 中的数字 `1`，第二个数组中数字`1`右边的下一个较大数字是 `3`。

对于 `num1` 中的数字 `2`，第二个数组中没有下一个更大的数字，因此输出 `-1`。

示例 2:

输入: `nums1 = [2,4]`, `nums2 = [1,2,3,4]`。

输出: `[3,-1]`

解释:

对于 `num1` 中的数字 `2`，第二个数组中的下一个较大数字是 `3`。

对于 `num1` 中的数字 `4`，第二个数组中没有下一个更大的数字，因此输出 `-1`。

提示:

`1 <= nums1.length <= nums2.length <= 1000`  
`0 <= nums1[i], nums2[i] <= 104`  
`nums1`和  
`nums2`中所有整数 互不相同  
`nums1` 中的所有整数同样出现在 `nums2` 中

## 单调栈解决(1)

解题思路:

1. 首先,我们把nums2逆序访问;
2. 判断 栈空 && nums2[i] 大于栈顶元素, 如果是, 则弹出栈中小于 nums2[i] 的元素;
3. 获取答案-->判断栈是否为空,栈为空代表该元素后面已经没有大于 nums2[i] 的值了,我们给答案变量num赋值为-1, 否则,栈不为空,将栈顶元素赋值给答案变量num(因为我们是单调递减栈,所以未被弹出的元素一定大于nums[i], 因此直接赋值即可);
4. 使用map容器保存 nums2[i] 和它对应的答案num;
5. nums2[i]入栈;
6. 循环结束后,我们的map容器已经存储了所有的值和答案,只需要把nums1中对应的值的答案放进数组ans即可;

```
class Solution {
public:
    vector<int> nextGreaterElement(vector<int>& nums1, vector<int>& nums2){
        stack<int> s;
        vector<int> ans;//答案数组
        int num;//答案变量
        map<int,int> map_ans;//存放值和相对应的答案

        for(int i=nums2.size()-1;i>=0;i--)//元素逆序入栈
        {
            while(!s.empty() && s.top()<= nums2[i])
            {
                s.pop();//弹出比该元素小的值
            }
            num=s.empty() ? -1 : s.top();//获取答案
            map_ans.insert(make_pair(nums2[i],num));//存答案
            s.push(nums2[i]);//当前元素入栈
        }

        for(int i=0;i<nums1.size();i++)
        {
            ans.push_back(map_ans[nums1[i]]);//将答案放进ans
        }
        return ans;
    }
};
```