496 下一个更大元素 I

Label：栈、数组

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 。

```
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 。
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 。
```

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

- 遍历

```java
class Solution {
    public int[] nextGreaterElement(int[] nums1, int[] nums2) {

        int[] re = new int[nums1.length];
        Arrays.fill(re, -1);

        for (int i = 0; i < nums1.length; i++) {

            int flag = 0;

            for (int j = 0; j < nums2.length; j++) {
               if (nums2[j] ==  nums1[i]) flag = 1;

               if (nums2[j] > nums1[i] && flag == 1) {
                    re[i] = nums2[j];
                    break;
               }
            }
        }
        return re;
    }
}
```

- 单调栈

```java
class Solution {
    public int[] nextGreaterElement(int[] nums1, int[] nums2) {
        int len1 = nums1.length;
        int len2 = nums2.length;

        Deque<Integer> stack = new ArrayDeque<>();
        Map<Integer, Integer> map = new HashMap<>(); // 记录右侧有大于值的所有结果
        // 先处理 nums2，把对应关系存入哈希表
        for (int i = 0; i < len2; i++) {
            while (!stack.isEmpty() && stack.peekLast() < nums2[i]) {
                map.put(stack.removeLast(), nums2[i]); // 有大于的，则更新
            }
            stack.addLast(nums2[i]);
        }

        // 遍历 nums1 得到结果集
        int[] res = new int[len1];
        for (int i = 0; i < len1; i++) {
            res[i] = map.getOrDefault(nums1[i], -1);
        }
        return res;
    }
}
```