

[每日 LeetCode] 496. Next Greater Element I

作者: [Hanseltu](#)

原文链接: <https://ld246.com/article/1556981395190>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

原文链接 [\[每日LeetCode\] 496. Next Greater Element I](#)

Description:

You are given two arrays (**without duplicates**) `nums1` and `nums2` where `nums1`'s elements are subset of `nums2`. Find all the next greater numbers for `nums1`'s elements in the corresponding places of `nums2`.

The Next Greater Number of a number `x` in `nums1` is the first greater number to its right in `nums2`. If it does not exist, output -1 for this number.

Example 1:

Input: `nums1 = [4,1,2]`, `nums2 = [1,3,4,2]`.

Output: `[-1,3,-1]`

Explanation:

For number 4 in the first array, you cannot find the next greater number for it in the second array, so output -1.

For number 1 in the first array, the next greater number for it in the second array is 3.

For number 2 in the first array, there is no next greater number for it in the second array, so output -1.

Example 2:

Input: `nums1 = [2,4]`, `nums2 = [1,2,3,4]`.

Output: `[3,-1]`

Explanation:

For number 2 in the first array, the next greater number for it in the second array is 3.

For number 4 in the first array, there is no next greater number for it in the second array, so output -1.

本题题意是给出两个数组，其中有一个是另一个的子数组，找出子数组中的每个元素在原数组中比这数大的第一个数。

思路一：暴力解法。依次遍历子数组中的元素，找到每个元素在原数组中的位置，然后在原数组中改的后面查找比此数大的数，有则返回第一个大的数，没有则返回-1。

思路二：使用`unordered_map`和`stack`结构，首先遍历原数组，将每个元素及此元素后面第一个大的组成键值对，然后遍历子数组，在`map`中查找键值，若存在则返回`value`值，不存在则返回-1。

C++代码) (思路一)

```
class Solution {
public:
    vector<int> nextGreaterElement(vector<int>& nums1, vector<int>& nums2) {
        vector<int> res(nums1.size());
        for (int i = 0; i < nums1.size(); ++i) {
            int j = 0, k = 0;
            for (; j < nums2.size(); ++j) {
                if (nums2[j] == nums1[i])
```

```

        break;
    }
    for (k = j + 1; k < nums2.size(); ++k) {
        if (nums2[k] > nums2[j]) {
            res[i] = nums2[k];
            break;
        }
    }
    if (k == nums2.size())
        res[i] = -1;
}
return res;
};

```

运行时间: 12ms

运行内存: 8.6M

C++代码 (思路二)

```

class Solution {
public:
    vector<int> nextGreaterElement(vector<int>& nums1, vector<int>& nums2) {
        stack<int> s;
        unordered_map<int, int> m;
        for (int n : nums2) {
            while (s.size() && s.top() < n) {
                m[s.top()] = n;
                s.pop();
            }
            s.push(n);
        }
        vector<int> ans;
        for (int n : nums1)
            ans.push_back(m.count(n) ? m[n] : -1);
        return ans;
    }
};

```

运行时间: 12ms

运行内存: 9.7M