Problem | Editorial | Submissions | Comments

Immediate Smaller Element

Basic | Accuracy: 36.26% | Submissions: 140K+ | Points: 1

Given an integer array **Arr** of size **N**. For each element in the array, check whether the right adjacent element (on the next immediate position) of the array is smaller. If next element is smaller, update the current index to that element. If not, then **-1**.

Example 1:

Input:
N = 5
Arr[] = {4, 2, 1, 5, 3}
Output:
2 1 -1 3 -1
Explanation: Array elements are 4, 2, 1, 5 3. Next to 4 is 2 which is smaller, so we print 2. Next of 2 is 1 which is smaller, so we print 1. Next of 1 is 5 which is

C++ (g++ 5.4) | Average Time: 20m | Start Timer

```
// } Driver Code Ends
//User function template for C++
class Solution{
public:
    void immediateSmaller(vector<int>&arr, int n) {
        //  code here
        for(int i = 0 ; i< n ; i++){
            if(arr[i] > arr[i+1]){
                arr[i] = arr[i+1];
            }else{
                arr[i] = -1;
            }
        }
        arr[n-1] = -1;
    }

};
// } Driver Code Ends
```

Custom Input | Compile & Run | Submit

Problem | Editorial | Submissions | Comments

Next Greater Element

Medium Accuracy: 32.95% Submissions: 297K+ Points: 4

Given an array **arr[ ]** of size **N** having elements, the task is to find the next greater element for each element of the array in order of their appearance in the array.

Next greater element of an element in the array is the nearest element on the right which is greater than the current element.

If there does not exist next greater of current element, then next greater element for current element is -1. For example, next greater of the last element is always -1.

Example 1:

Input:
N = 4, arr[] = [1 3 2 4]
Output:
3 4 4 -1
Explanation:

C++ (g++ 5.4) | Average Time: 20m | Start Timer

```cpp
// } Driver Code Ends
class Solution
{
    public:
    //Function to find the next greater element for each element of the array.
    vector<long long> nextLargerElement(vector<long long> arr, int n){
        // Your code here
        vector<long long>res(n);
        stack<long long int>s;
        s.push(-1);

        for(int i =n-1 ; i>=0 ;i--){
         while(s.top()!=-1 && s.top()<=arr[i]){
             s.pop();
         }
         res[i] = s.top();
         s.push(arr[i]);
        }

        return res;
    }
};
// } Driver Code Ends
```

Custom Input | Compile & Run | Submit

Problem | Submissions | Comments

Next Greater Element 2

Medium | Accuracy: 67.89% | Submissions: 2K+ | Points: 4

Given a circular interger array **arr** of size **N** (i.e ., the next element of **arr [N-1]** is **arr[0]**), return the **next greater number** for every element in **arr**.

The **next greater element** of a number **x** is the **first greater number** to its traversing-order next in the array, which means you could search circularly to find its next greater number. If it doesn't exist, return **-1** for this number.

Example 1:

Input:
N = 3
arr[ ] = {1, 2, 1}
Output: {2, -1, 2}
Explanation: The first 1's next greater number is 2:

C++ (g++ 5.4) | Average Time: 30m | Start Timer

```
// } Driver Code Ends
class Solution {
  public:
    vector<int> nextGreaterElement(int n, vector<int>& arr) {
        // code here
        vector<int> res;
        stack<int> s;
        for(int i = 2*n -1 ; i>= 0 ; i--){
            while(!s.empty() && s.top() <= arr[i%n]){
                s.pop();
            }
            if(i < n){
                if(!s.empty()){
                    res.push_back(s.top());
                }else{
                    res.push_back(-1);
                }
            }
            s.push(arr[i%n]);

        }
        reverse(res.begin() , res.end());
        return res;
    }
};
// } Driver Code Ends
```

Custom Input | Compile & Run | Submit