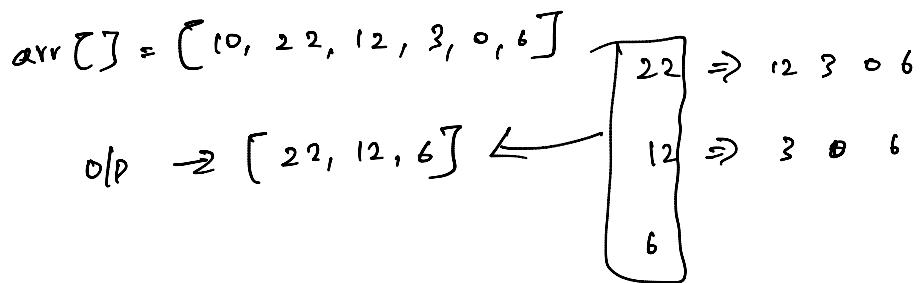


26. Leaders in an Array

Leaders in an Array: Everything on the right should be smaller. Last element is always leader.



1) Brute force:

↓
[10, 22, 12, 3, 0, 6]

\rightarrow Take 10, & check on right if anything greater than 10 then it won't be leader.

Same for every element

\rightarrow for every single element do a linear search on the right portion of the array.

```
for(i=0 ; i<n ; i++)  
{  
    leader = true;  
    for(j=i+1 ; j<n ; j++)  
        if (a[i] > a[j])
```

```

    if [a[i] > a[i])
        leader = false;
        break;
    }
}

if (leader == true)
    ans.add(a[i])
}

```

$T.C \Rightarrow O(n^2)$ $S.C \Rightarrow O(N)$ (for storing the ans in the worst case)
else $O(1)$

2) Optimal Solution

$$\text{arr}[] = \{10, 22, 12, 3, 0, 6\}$$

→ If the element is 12, whatever is the maxi element on the right which is 6,
if $12 > 6$, which means 12 is greater than every element on the right.

→ if the el is 22, the maxi element on right is 12
and $22 > 12$, so 22 is greater than every element.

\Rightarrow If array element \geq maxi on the right

then arr element will be the leader.

\rightarrow So start iterating from back.

- maxi = ~~INT_MIN~~; arr[7] = [10, 22, 12, 3, 0, 6]
12
6 is always greater because last element
 $0 \geq 6 \times$ 22? 12 ✓
 $3 \geq 6 \times$ 12? 6 ✓
 $10 \geq 12 \times$
return 10

\rightarrow Since we did back traversal, element will be stored in \leftarrow way.

\rightarrow if the problem says to return in the original array format. (22, 12, 6)

\rightarrow Then you can just reverse the ans.

\rightarrow If they ask in sort order then do sorting, $O(N \log N)$ for this

Problem Statement Suggest Edit

There is an integer array 'A' of size 'N'.
An element is called a Superior Element if it is greater than all the elements present to its right.
You must return a sorted array of all Superior Elements in the array 'A'.
Note:
The last element of the array is always a Superior Element.

For example

Input:
A = [1, 2, 3, 2], N = 4
Output:
2 3
Explanation:
A[2] = 3 is greater than A[3]. Hence it is a Superior Element.
A[3] = 2 is the last element. Hence it is a Superior Element.
The final answer is in sorted order.

```

1 vector<int> superiorElements(vector<int>&a) {
2     vector<int> ans;
3     int maxi = INT_MIN;
4     int n = a.size();
5     // O(N)
6     for(int i = n-1; i>=0; i--){
7         if(a[i] > maxi){
8             ans.push_back(a[i]);
9         }
10        //keep track of right max
11        maxi = max(maxi, a[i]);
12    }
13    // O(NlogN)
14    // sort(ans.begin(), ans.end());
15    return ans;
16 }

```

No need to sort it as
it will store answer in the
sorted order when taken from
back (right \rightarrow left)

GFG Solution

```

11* class Solution{
12     //Function to find the leaders in the array.
13     public:
14     vector<int> leaders(int a[], int n){
15         vector<int> ans;
16         int maxi = INT_MIN;
17         for(int i = n-1; i>=0; i--){
18             if(a[i] > maxi){
19                 ans.push_back(a[i]);
20             }
21             //keep track of right max
22             maxi = max(maxi, a[i]);
23         }
24         reverse(ans.begin(), ans.end());
25         return ans;
26     };
27 }
28 // } Driver Code Ends

```

\Rightarrow Here as per problem statement, just use
reverse fn.