

NEXT GREATER ELEMENT

Part of Monotonic Stack Problems

When elements are stored in a specific order

Problem states that for every element in the array, we are supposed to return an array containing the next greater element. Eg:

arr :

6	0	8	1	3
---	---	---	---	---

Ans :

8	8	-1	3	-1
---	---	----	---	----

Brute force solution is to iterate through the array, and for each element iterate again to find next greatest. But time complexity is $O(N^2)$

For better solution, we take a stack, start from the right of the array and keep adding them to the stack if current element is lesser than top element (\because top element is your next greater). But if its greater, we pop the stack until we find the next

greater element.

Only Run:

4	12	5	3	1	2	5	3	1	2
---	----	---	---	---	---	---	---	---	---

Step 1:

↑
-1

2
-1

4	12	5	3	1	2	5	3	1	2
---	----	---	---	---	---	---	---	---	---

↑ X
2 -1

1
2
-1

4	12	5	3	1	2	5	3	1	2
---	----	---	---	---	---	---	---	---	---

↑ X X
-1 2 -1

3
-1

4	12	5	3	1	2	5	3	1	2
---	----	---	---	---	---	---	---	---	---

↑ X X X
-1 -1 2 -1

5
-1

4	12	5	3	1	2	5	3	1	2
---	----	---	---	---	---	---	---	---	---

↑ X X X X
5 -1 -1 2 -1

2
5
-1

4	12	5	3	1	2	5	3	1	2
---	----	---	---	---	---	---	---	---	---

↑ X X X X X
2 5 -1 -1 2 -1

1
2
5
-1

4	12	5	3	1	2	5	3	1	2
---	----	---	---	---	---	---	---	---	---

↑ ~~X~~ ~~X~~ ~~X~~ ~~X~~ ~~X~~ ~~X~~ ~~X~~
 5 2 5 -1 -1 2 -1

X
X
5

-1

4	12	5	3	1	2	5	3	1	2
---	----	---	---	---	---	---	---	---	---

↑ ~~X~~ ~~X~~ ~~X~~ ~~X~~ ~~X~~ ~~X~~ ~~X~~
 -1 5 2 5 -1 -1 2 -1

5

-1

4	12	5	3	1	2	5	3	1	2
---	----	---	---	---	---	---	---	---	---

↑ ~~X~~ ~~X~~ ~~X~~ ~~X~~ ~~X~~ ~~X~~ ~~X~~
 -1 -1 5 2 5 -1 -1 2 -1

12

-1

4	12	5	3	1	2	5	3	1	2
---	----	---	---	---	---	---	---	---	---

↑ ~~X~~ ~~X~~ ~~X~~ ~~X~~ ~~X~~ ~~X~~ ~~X~~
 12 -1 -1 5 2 5 -1 -1 2 -1

4
12

-1

End

We have our Next Greater Element Array

Pseudocode :

```
nextGreaterElement (arr) {
  stack <int> st;
  for (int i = arr.size(); i >= 0; i--) {
    if (st.empty())
      st.push (arr[i]);
    else
      st.push (arr[i] = st.top());
```

```
    ans[i] = -1;
}
else {
    if (arr[i] < st.top()) {
        ans[i] = st.top();
        st.push(arr[i]);
    }
    else {
        while (arr[i] > st.top()) 88  
!st.empty()
            st.pop();
        ans[i] = st.top();
        st.push(arr[i]);
    }
}
return ans;
```