Q1 Given an array. Find the nearest smaller element on left for every element.

↳ Compared to the element you are finding for (

Ex1

A[] :	4	5	2	10	11	2
NSE :	-1	4	-1	2	10	-1

Ex2

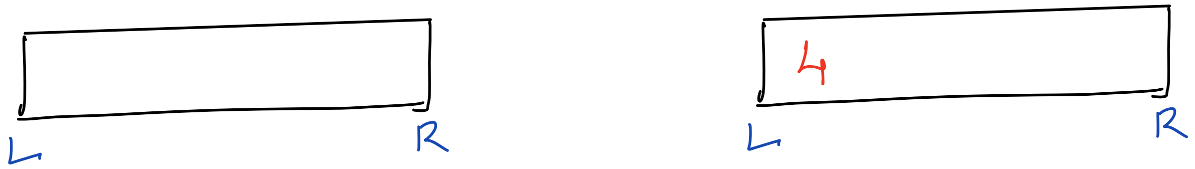
A[] :	4	6	10	11	7	8	3	5
NSE :	-1	4	6	10	6	7	-1	3

Brute force : TC: $O(n^2)$. Nested loop !!

4 6 10 11 7 8 3 5

-1 4

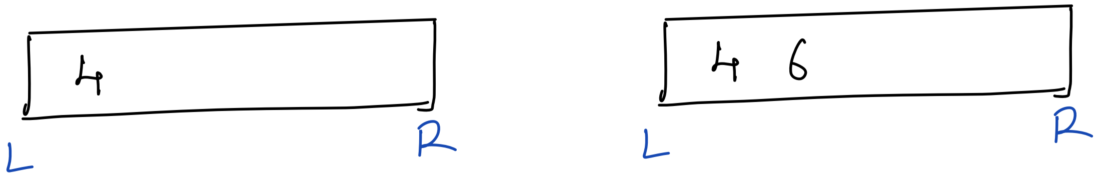
Processing 4 = -1



4 6 10 11 7 8 3 5

-1

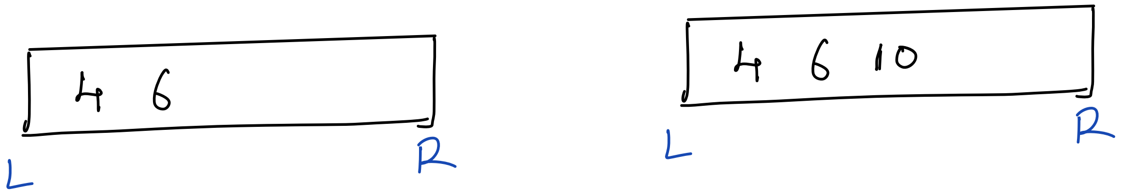
Processing 6 = 4



4 6 10 11 7 8 3 5

-1

Processing 10 = 6



4	6	10	11	7	8	3	5
-1							

Processing 11 = 10

4 6 10
L R

4 6 10 11
L R

4	6	10	11	7	8	3	5
-1	4	6	10	6			

Processing 7 = 6

4 6 ~~10~~ ~~11~~
L R

=

4 6 7
L R

4	6	10	11	7	8	3	5
-1	4	6	10	6	7		

Processing 8 = 7

4 6 7
L R

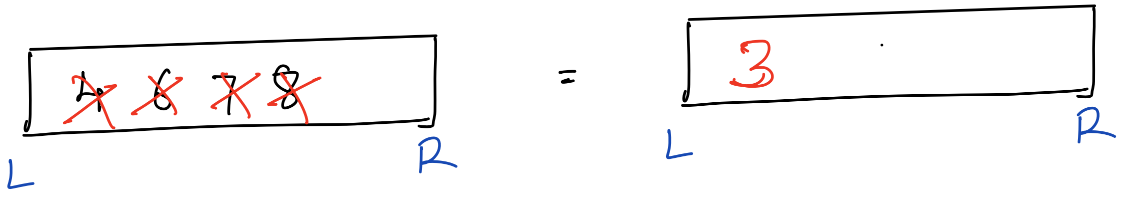
=

4 6 7 8
L R

4 6 10 11 7 8 3 5

-1 4 6 10 6 7 -1

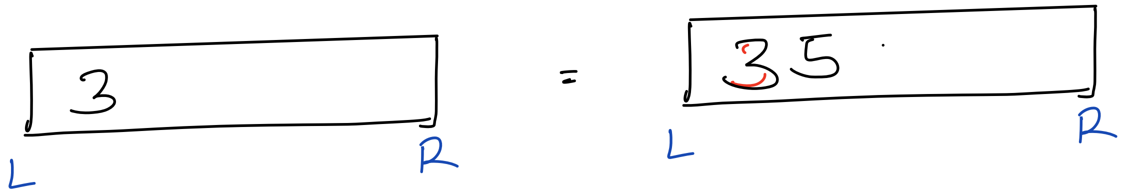
Processing 3 = -1



4 6 10 11 7 8 3 5

-1 4 6 10 6 7 -1 3

Processing 5 = 3



Time Complexity : $n + n \approx O(n)$

iterating every element

At max n removals.

Pseudo Code

```
stack<int> s;   int nsl[n];

for (int i=0 ; i<n ; i++) {
    while ( !st.empty () && st.top() ≥ arr[i])
          st.pop();

    if (st.empty())
          nsl[i] = -1;
    else
          nsl[i] = st.top();

    st.push(arr[i])
}
```

$>$ $\le$ $<$

1) nearest smaller or equal to left.

2) nearest greator to left.

3) nearest greator or equal to left

Q2 Given an array. Find the nearest smaller element on right for every element.

↳ Compared to the element you are finding for (

Pseudo Code

```
stack<int> s;   int nsr[n];

for (int i=(n-1) ; i>=0 ; i--) {
|    while ( !st.empty () && st.top() >= arr[i])
|           st.pop();
|
|    if (st.empty())
|         nsr[i] = -1;
|    else
|         nsr[i] = st.top();
|
|    st.push(arr[i])
|
}
```

Q3 Find the index of the NSL.

Ex1

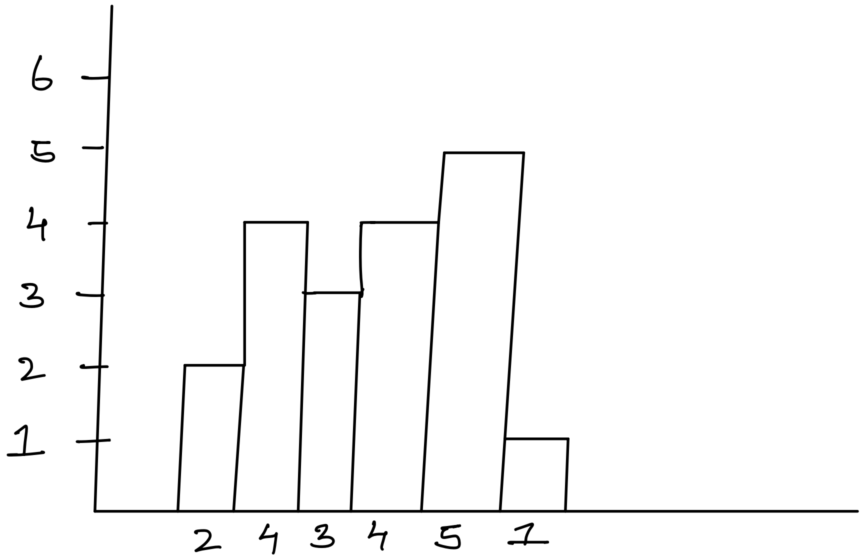
	0	1	2	3	4	5	6	7
arr[] =	4	6	10	11	7	8	3	5
NSL =	-1	4	6	10	6	7	-1	3
nsli =	-1	0	1	2	1	4	-1	6

Pseudo Code

```
stack<int> s;   int nsli [n];

for (int i=0 ; i<n ; i++) {
    while (!st.empty () && arr[st.top()] ≥ arr[i])
            st.pop();

    if (st.empty())
            nsl[i] = -1;
    else
         nsl[i] = st.top();

    st.push( i )
}
```

Q4 Given a histogram. Find the max rectangular area contained in it!

Ex1 : A[] : 2 4 3 4 5 1



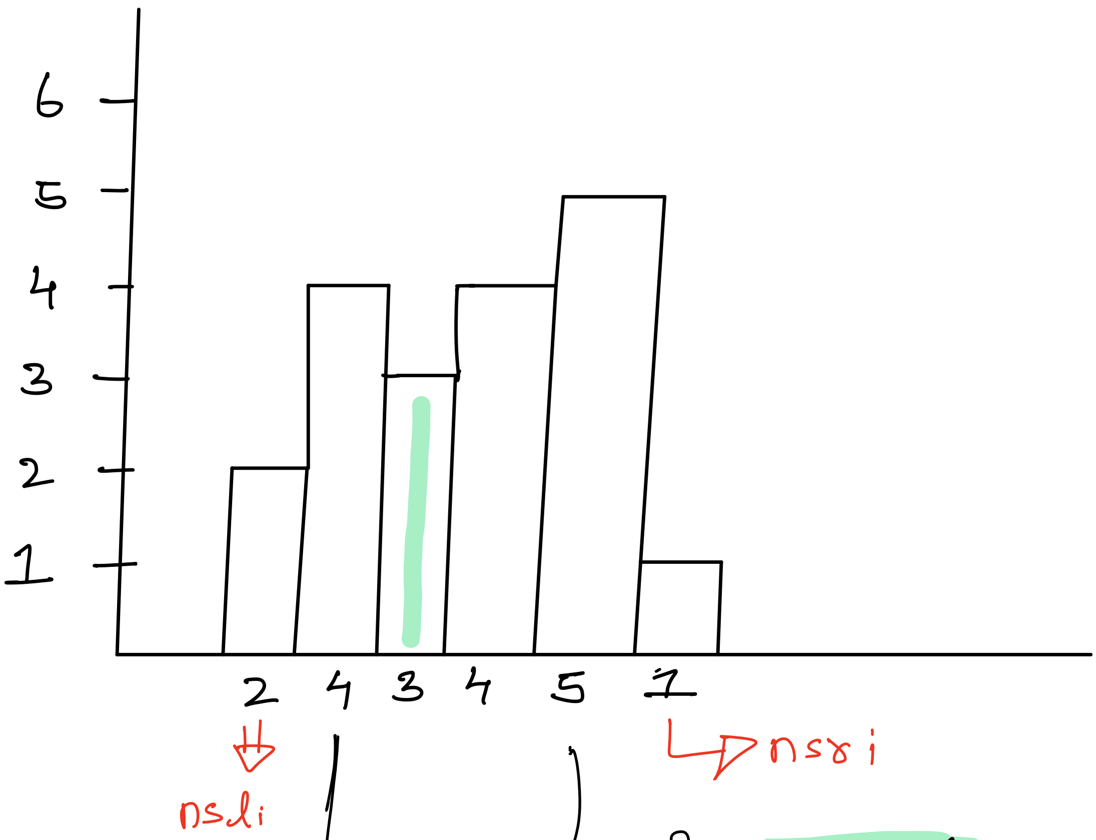
Brute Force :

1) Using 2 pointers, you can fix 2 ends of a rectangle. This gives length

2) Traverse all bars in between. Find Smallest. This height.

3) Find Area = length x height. Update max_ans.

TC: O(n^3)
SC: O(1).



P_2 = nsri − 1

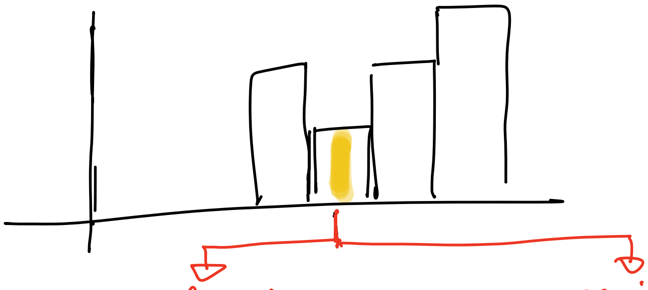
P_1 = nsli + 1

Length => $(P_2 - P_1 + 1)$

Height => arr[i]

Area = Length x Height

EDGE CASE !



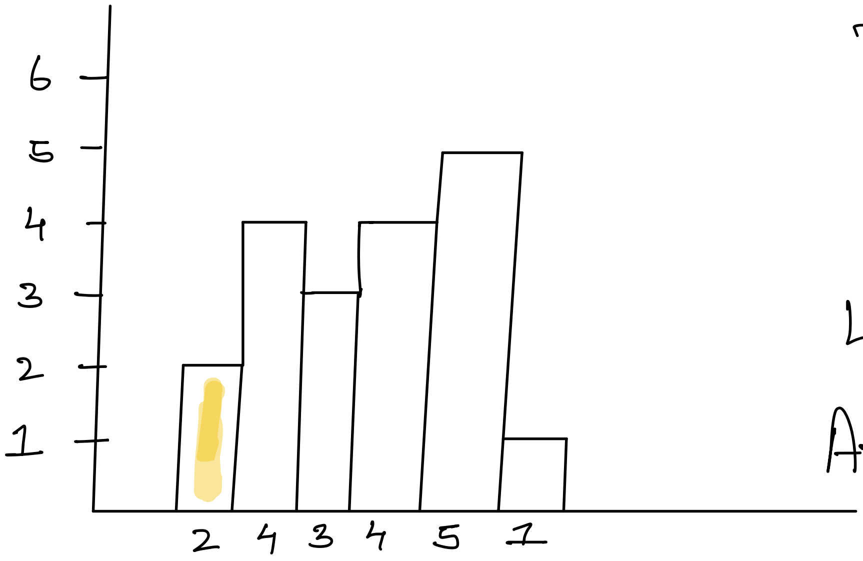
nsli = −1

P_1 => 0

nsri = −1

P_2 = n−1

		0	1	2	3	4	5
A[]	:	2	4	3	4	5	1
nsli	:	-1	0	0	2	3	-1
nsri	:	5	2	5	5	5	-1



$P_1 \Rightarrow 0$

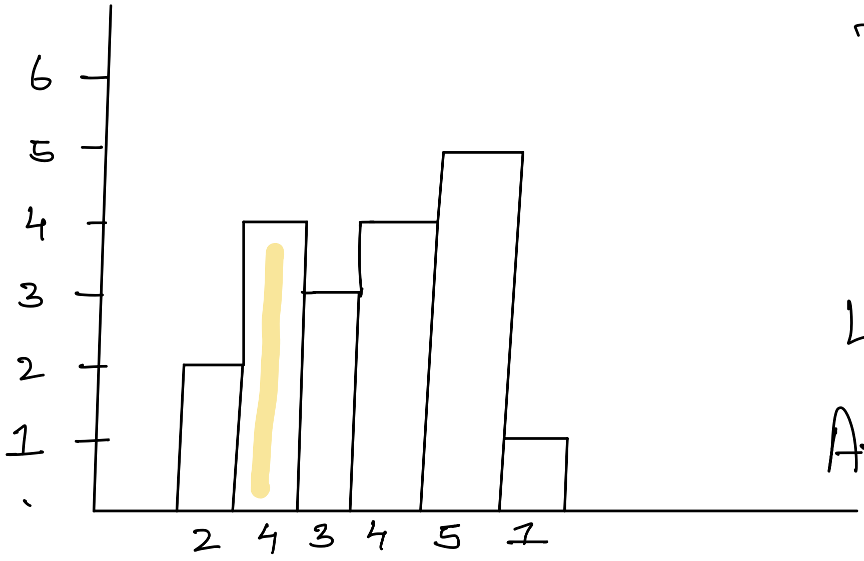
$P_2 \Rightarrow 4$

$h \Rightarrow 2$

$L \Rightarrow 4-0+1$
$= 5$

Area = 10

		0	1	2	3	4	5
A[]	:	2	4	3	4	5	1
nsli	:	-1	0	0	2	3	-1
nsri	:	5	2	5	5	5	-1



$P_1 \Rightarrow 1$

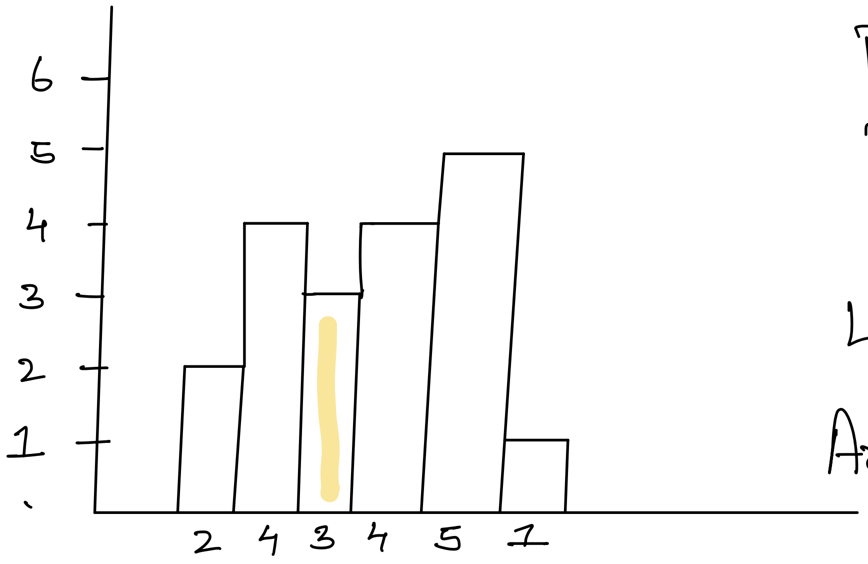
$P_2 \Rightarrow 1$

$h \Rightarrow 4$

$L \Rightarrow 1-1+1$
$= 1$

Area = 1×4 = 4

	0	1	2	3	4	5
A[] :	2	4	3	4	5	1
nsli :	-1	0	0	2	3	-1
nsri :	5	2	5	5	5	-1



$P_1 \Rightarrow 1$

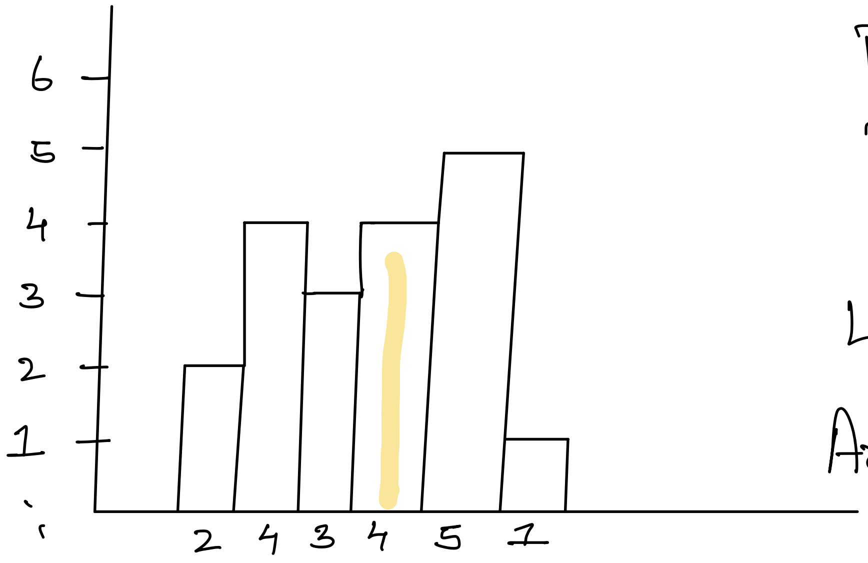
$P_2 \Rightarrow 4$

$h \Rightarrow 3$

$L \Rightarrow 4-1+1=4$

Area $= 4\times3 = 12$

	0	1	2	3	4	5
A[] :	2	4	3	4	5	1
nsli :	-1	0	0	2	3	-1
nsri :	5	2	5	5	5	-1



$P_1 \Rightarrow 3$

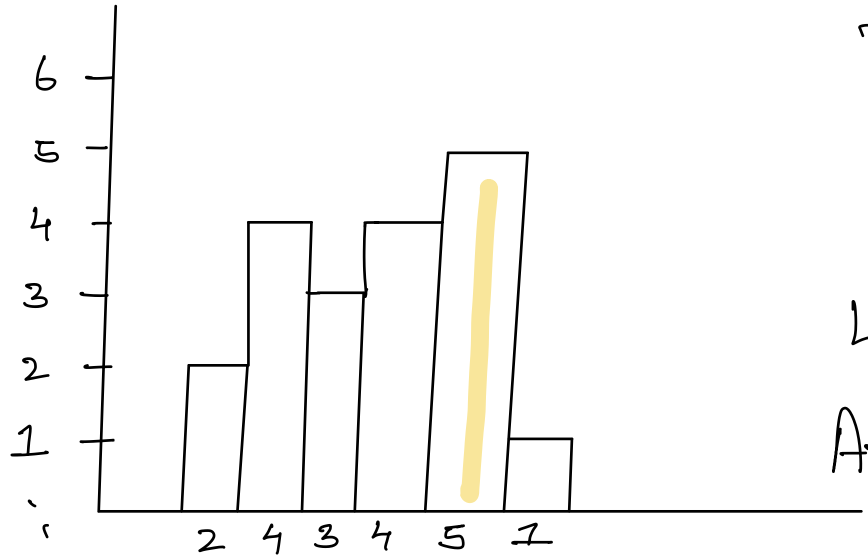
$P_2 \Rightarrow 4$

$h \Rightarrow 4$

$L \Rightarrow 4-3+1=2$

Area $= 4\times2 = 8$

		0	1	2	3	4	5
A[]	:	2	4	3	4	5	1
nsli	:	-1	0	0	2	3	-1
nsri	:	5	2	5	5	5	-1



$P_1 \Rightarrow 4$

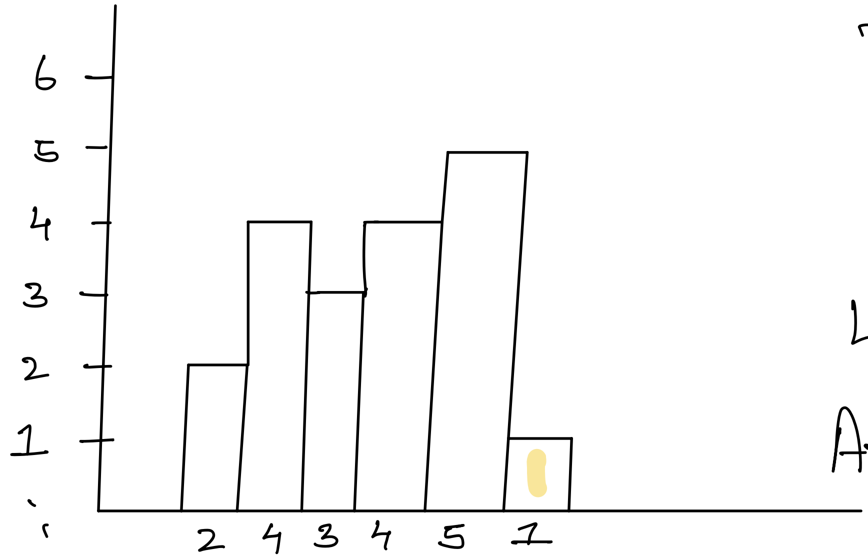
$P_2 \Rightarrow 4$

$h \Rightarrow 5$

$L \Rightarrow 4-4+1 = 1$

$\text{Area} = 5 \times 1 = 5$

		0	1	2	3	4	5
A[]	:	2	4	3	4	5	1
nsli	:	-1	0	0	2	3	-1
nsri	:	5	2	5	5	5	-1



$P_1 \Rightarrow 0$

$P_2 \Rightarrow 5$

$h \Rightarrow 1$

$L \Rightarrow 5-0+1 = 6$

$\text{Area} = 1 \times 6 = 6$

Pseudo Code

```
1) Calculate nsli [n]
2) Calculate nsri [n]

 ans = Int.Min;
for (int i=0; i<n; i++) {
      if (nsli [i] == -1)
            $P_1 = 0$;
      else
            $P_1 = nsli [i] + 1$;

      if (nsri [i] == -1)
            $P_2 = n - 1$

      else
            $P_2 = nsri [i] - 1$

      $len = P_2 - P_1 + 1$;

      ans = max (ans, len * arr [i]);

}

  return ans;
```

TC: O(n)
SC: O(n)

Q5 Given an integer array with distinct integers. ∀ subarrays, Find (max - min) & return its sum as the answer.

Ex1: $A[3] : \begin{array}{ccc} 0 & 1 & 2 \\ \hline 2 & 5 & 3 \end{array}$

Subarray	Max	Min	Difference.
[2]	2	2	0
[2 5]	5	2	3
[2 5 3]	5	2	3
[5]	5	5	0
[5 3]	5	3	2
[3]	3	3	0
			8

$$4(5) = 1(5) + 1(2) - 3(2) + 1(3) - 2(3)$$

$$20 - 5 + 2 - 6 + 3 - 6$$

$$\Rightarrow 8$$

Contribution of A[i]

$$A[i] \times \left\{ \text{no of times } A[i] \text{ is max in a subarray} \;-\; \text{no of times } A[i] \text{ is min in a subarray} \right)$$

no of subarrays where A[i] is maximum.

$$\left[\begin{array}{ccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 8 & 3 & 5 & 4 & 2 & 11 & 7 & 2 \end{array}\right]$$

Starting points : [3, 5]

Ending points : [5, 4, 2]

Total subarrays ⇒ $2 \times 3 = 6$

$$\begin{array}{ccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \end{array}$$

$$\left[\begin{array}{ccccccccc} 1 & 8 & 3 & 5 & 4 & 2 & 11 & 7 & 2 \end{array}\right]$$

ngri

ngli $P_1 \Rightarrow ngli + 1$

$P_2 = ngri - 1$

no of starting points $\Rightarrow [P_1, i]$

$\downarrow$

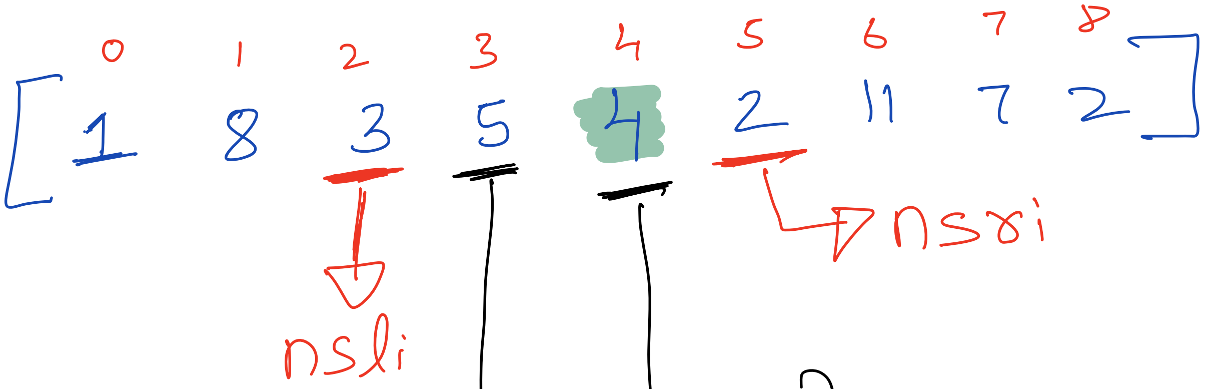
$i - P_1 + 1$

no of ending points $= [i, P_2]$

$\downarrow$

$(P_2 - i + 1)$

no of subarrays where A[i] is minimum!



$P_1 = nsli + 1$ $\qquad$ $P_2 = nsri - 1$

Starting points $= [P_1, i]$

$= i - P_1 + 1$

Ending Points $= [i, P_2]$

$= P_2 - i + 1$

Time Complexity!

1) Precompute nsli, nsri, ngli, ngri

$\Rightarrow O(n)$

2) Traversing each element $\Rightarrow O(n)$

TC: $O(n)$ $\qquad$ SC: $O(n)$