Welcome ☺

Q Given an integer array A, find the index of nearest smaller element on left +i index of A[]. Formally, +i find j st A[j] < A[i] & j<i & j has to be max.

eg:

0	1	2	3	4	5	6	7
8	2	4	9	7	5	3	10
-1	-1	1	2	2	2	1	6

Sol 1 ⟹ Bruteforce ⟹ ∀index travel towards left & find nearest smaller element.

$$T.C \to O(N^2)$$
$$S.C \to O(1)$$

Sol 2

0	1	2	3	4	5	6	7
8	_	_	_	_	5	$\underline{x}$	_

$x \leq 5$ Ans != 0 ⎫
$5 < x \leq 8$ Ans = 5   ⎬   Ans != 0
$x > 8$ Ans = 5   ⎭

0	1	2	3	4	5	6	7
8	2	4	9	7	5	3	10
-1	-1	1	2	2	2	1	6

↳ check for the rightmost (latest) smaller element on left. → stack

Code
```
for ( i -> 0 to N-1)
{
    while ( ! st.isEmpty() && A[st.peek()] >= A[i])
    {
        st.pop()
    }
    if ( st.isEmpty())
        ans[i] = -1
    else
        ans[i] = st.peek()

    st.push(i)      T.C -> O(N)
}                   S.C -> O(N)
```

```
 0    1    2    3    4    5    6    7
 8    2    4    9    7    5    3    10
-1   -1    1    2    2
```

Stack:
```
4
3
2
1
0
```

Q2 At i find nearest smaller or equal on left

Q3 At i find nearest greater on left

Q4 At i find nearest greater or equal on left

Q5 At i find nearest smaller on right

Code
```
for ( i -> N-1 to 0)   R -> L
{
    while ( ! st.isEmpty() && A[st.peek()] >= A[i])
    {
        st.pop()
    }
    if ( st.isEmpty())
```

(>=   <=   <)

ans [i] = -1
            else
                    ans [i] = st.peek()

            st.push (i)   T.C → O(N)
        }   S.C → O(N)

    ✓   ✓   ✓   ✓   ✓
    0   1   2   3   4
    8   2   4   9   7   5   3   10
   -1 -1   1   2   2

                                              4
                                              3
                                              2
                                              1
                                              0

Q6 Ai find nearest smaller or equal on right
Q7 Ai find nearest greater on right
Q8 Ai find nearest greater or equal on right

Q9   Given an integer array with distinct integers,
     ∀subarrays, find (max - min) & return its sum as ans.

     eg: [2   5   3]   ⇒   2 ⇒ (2-2)   5 ⇒(5-5)
                                   2 5 ⇒ (5-2)   5 3 (5-3)
                                   2 5 3 ⇒ (5-2) 3   (3-3)
                                                    8

Sol1 Bruteforce

       ∀subarray   T.C ⇒ O(N³) ⟶ O(N²)
       Σ (max-min)   S.C ⇒ O(1)

Contribuⁿ Tech

starting- end

eg: [1   8   3   5   4   2   11   7   12]

0   1   2   3   4   5   6   7   8

j   i   k
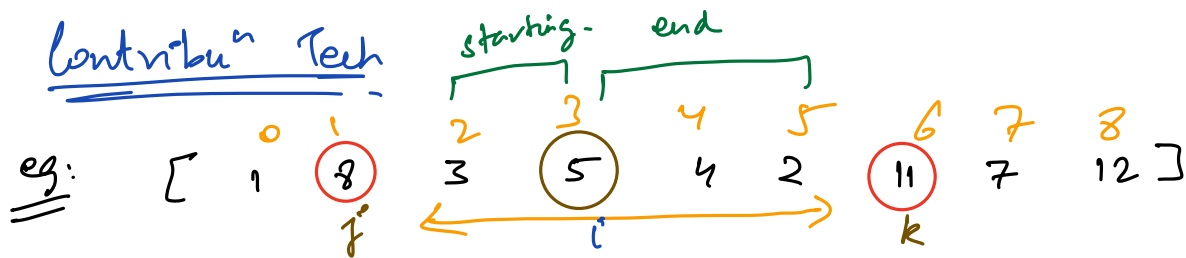
$$\sum_{i=0}^{N-1} \text{ Contribution of } A[i]$$

$$\text{Contribuⁿ} = A[i] * \left(\begin{array}{l} \text{\# subarrays where} \\ A[i] \text{ is max} \end{array} - \begin{array}{l} \text{\# subarrays} \\ \text{where } A[i] \\ \text{is minimum} \end{array} \right)$$

j → nearest greater on left of i

k → nearest greater on right of i

\# subarrays ⟹ $(i-j) * (k-i)$

Code

$$\text{Ans} = \sum_{i=0}^{N-1} A[i] * \left(\begin{array}{l} (i - ng\_left(i)) * (ng\_right(i) - i) \\ \rule{2cm}{0.4pt} \text{ minus} \\ (i - ns\_left(i)) * (ns\_right(i) - i) \end{array} \right)$$

T.C → $O(N)$

S.C → $O(N)$

Q10 Given an integer array A, where A[i] is the height of iᵗʰ bar. width of each bar is 1. Find the area of largest rectangle formed by continuous bars.

eg.

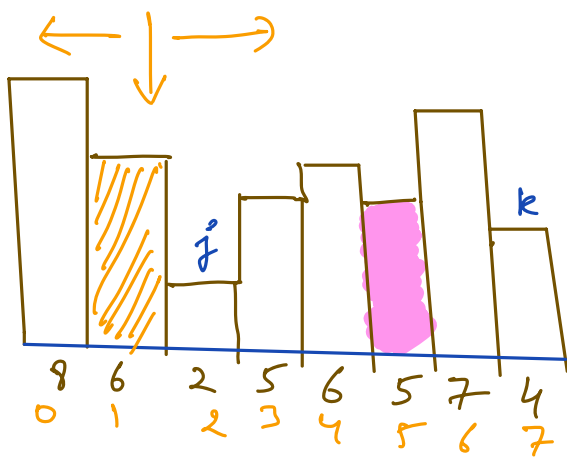$6 * 2 = 12$

$4 * 5 = \boxed{20}$

$2 * 7 = 14$

8	6	2	5	6	5	7	4
0	1	2	3	4	5	6	7
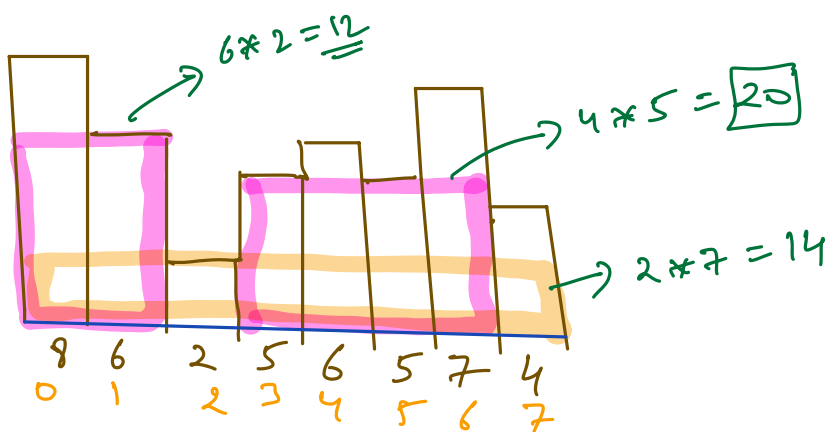
Brute force

∀ subarrays as base
find min height & calc. area

min ele. in subarray

B*H

T.C → $O(N^3)$ → $O(N^2)$

S.C → $O(1)$

8	6	2	5	6	5	7	4
0	1	2	3	4	5	6	7

j → smaller ele. on left of i | k → nearest smaller ele. on right of i

$$Area = (k - j - 1) * A[i]$$

→ if not present return N

Code

$$\forall_{i=0}^{N-1} \ max \left(A[i] * \left(ns\_right(i) - ns\_left(i) - 1 \right) \right)$$

if not present return -1

T.C → $O(N)$

S.C → $O(N)$