

Q.1

Nearest Smallest Element

Google
Facebook
Amazon
MS/Adobe
.....

Given an array of +ve integers.

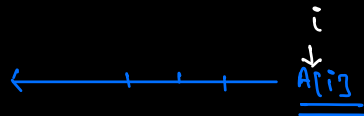
For every i , find the nearest element on the left side of i which is smaller than $A[i]$.

	0	1	2	3	4	5
A:	4	2	5	10	8	2
	↓	↓	↓	↓	↓	↓
	-1	-1	2	5	5	-1

Quiz

	0	1	2	3	4	5	6	7
A:	4	6	10	11	7	8	3	5
	↓	↓	↓	↓	↓	↓	↓	↓
	-1	4	6	10	6	7	-1	3

Brute force:



$ans[0] = -1;$

for every index i :

iterate from j : $i-1$ to 0 :

check for ele smaller than $A[i]$.

TC: $O(N^2)$

SC: $O(1)$

↳ Including the $ans[]$ array.

A: 4 6 2 8 6 1

ans: -1 4 -1 2 2 -1

Box

~~4~~ ~~6~~ ~~2~~ ~~8~~ ~~6~~ 1

→ Stack.

A: 4 5 2 10 18 \downarrow 2

ans: -1 4 -1 2 10 -1

2
~~18~~
~~10~~
2
~~5~~
~~4~~

↑
increasing
order.

Code

```
int ans[N];  
ans[0] = -1;  
stack<int> st;  
st.push(A[0]);  
for(i = 0; i < N; i++) {  
    while(!st.empty() && st.top() >= A[i])  
        st.pop();  
    if(st.empty()) {  
        ans[i] = -1;  
    }  
    else {  
        ans[i] = st.top();  
    }  
    st.push(A[i]);  
}
```

TC: $O(N)$ { Max : $2N$ iterations }

SC: $O(N)$

↓
Stack.

$A: \begin{matrix} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ & 4 & 6 & 10 & 11 & 7 & 8 & 3 & 5 \end{matrix}$
 $\text{ans: } \begin{matrix} & -1 & 4 & 6 & 10 & 6 & 7 & -1 & 3 \end{matrix}$

$\begin{matrix} 5 \\ 3 \\ 8 \\ 4 \\ 11 \\ 10 \\ 6 \\ 4 \end{matrix}$

Q.2 Find the index of Nearest smaller Element.

```

int ans[N];
stack<int> st;
for(i=0; i<N; i++){
    while(!st.empty() && A[st.top()] >= A[i])
        st.pop();
    if(st.empty()){
        ans[i] = -1;
    }
    else{
        ans[i] = st.top();
    }
    st.push(i);
}

```

3

Q.3 Get the distance of Nearest smaller element on left side.

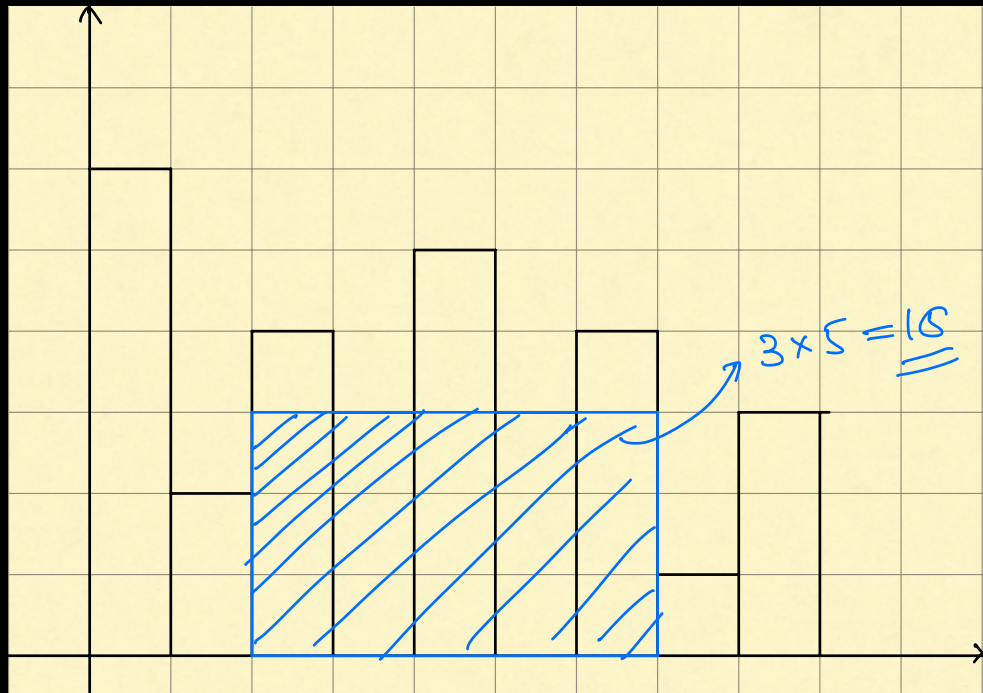
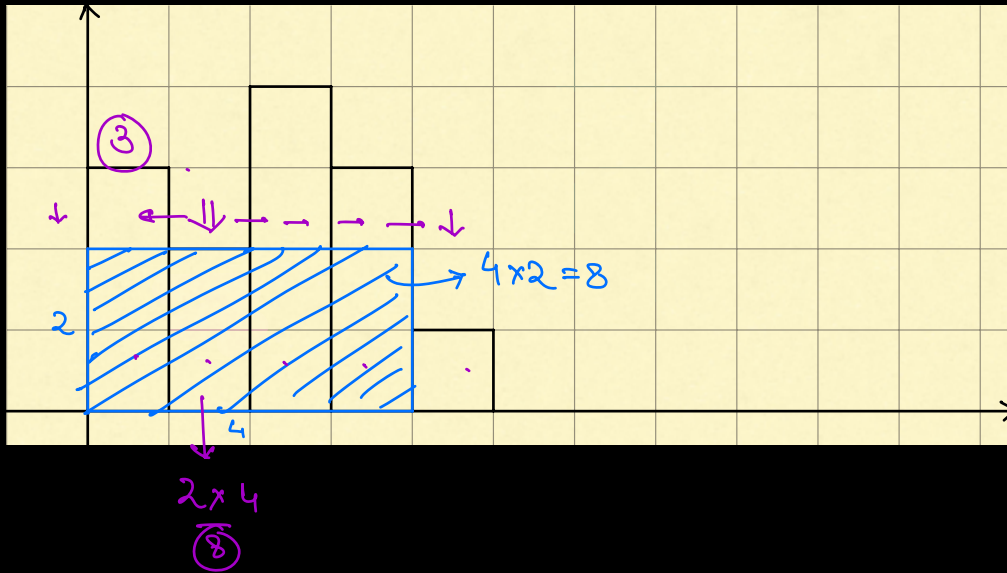
Q.4 Nearest smaller element on the right side.
($N-1 \rightarrow 0$)

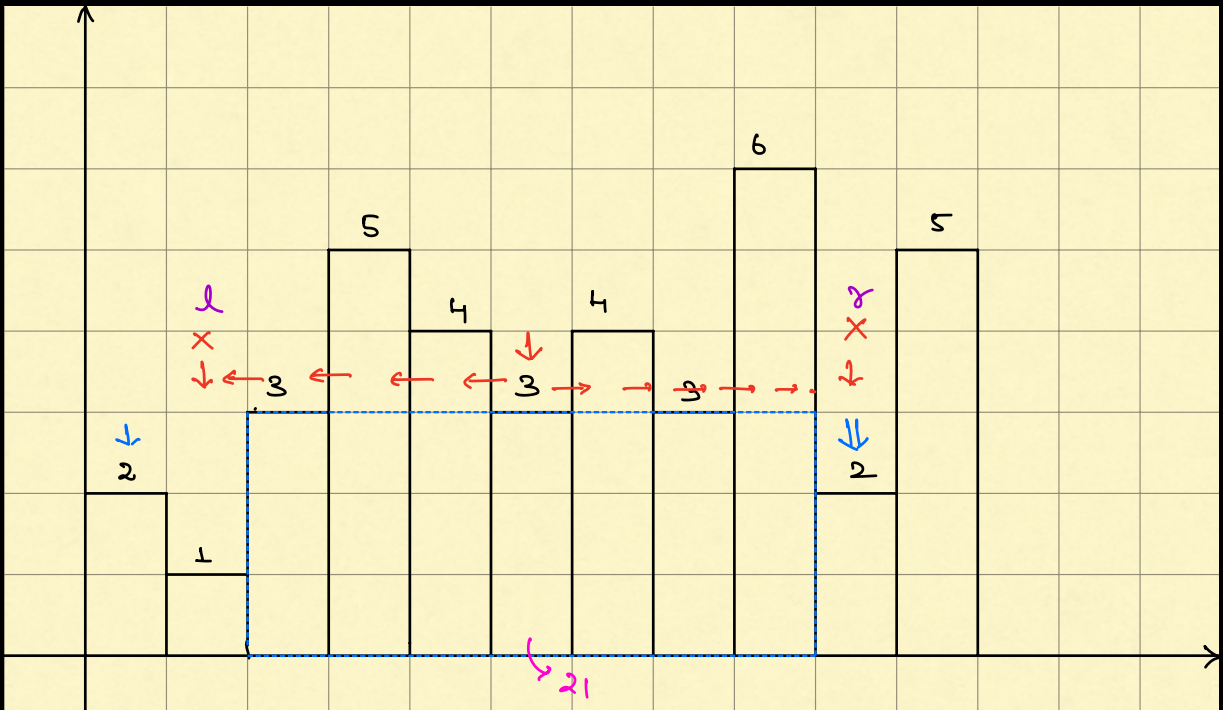
Q.5 Get the distance of Nearest smaller element on right side.

Q.6 Nearest greater on the left side.

Q.7 Nearest greater on the right side.

Q.8 Largest rectangle Area in Histogram.





$A: [2, 1, 3, 5, 4, 3, 4, 3, 6, 2, 5]$
 $\quad \quad \quad \uparrow \quad \quad \quad \uparrow$
 $\quad \quad \quad \text{NSL} \quad \quad \quad \text{NSR}$

$$l = 1, r = 9 \Rightarrow (l, r)$$

$$\begin{aligned}
 w &= r - l - 1 \\
 &= 9 - 1 - 1 = 7
 \end{aligned}$$

$$h = 3.$$

$$A = w \times h$$

$$= 21$$

NSL[i]: index of Nearest Smaller Element on left of i ;

NSR[i]: index of Nearest Smaller Element on right of i .

// NSL[] : indices $\Leftarrow O(N)$

// NSR[] $\Leftarrow O(N)$

$O(N)$ { for($i=0$; $i < N$; $i++$) {
 $l \leftarrow \text{NSL}[i]$
 $r \leftarrow \text{NSR}[i]$
 $w = r - l - 1$;
 $\text{area} = \max(\text{area}, w \times A[i])$;
}

Tc: $O(N)$

Sc: $O(N)$

Q. Given an Array.
Google. Find the sum of $(\text{max} - \text{min})$ for all possible subarrays.
MS.

A: $\overset{0}{2} \quad \overset{1}{5} \quad \overset{2}{3}$

s	e	max	min	max-min
0	0	2	2	0
0	1	5	2	3
0	2	5	2	3
1	1	5	5	0
1	2	5	3	2
2	2	3	3	0

8

$$2 \times 1 - 2 \times 3 + 5 \times 4 - 5 \times 1 + 3 \times 1 - 3 \times 2 \Rightarrow$$

Brute force

for each subarray, find MAX & MIN

$$\underline{\underline{TC: O(N^3)}}$$

$$\frac{N(N+1)}{2} : \text{No. of subarrays.}$$

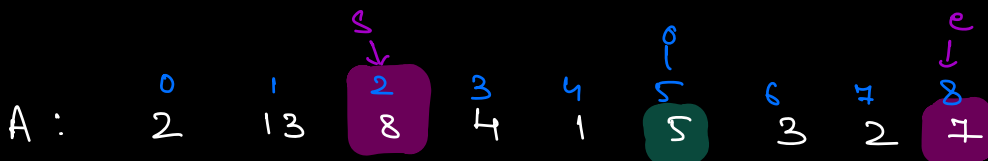
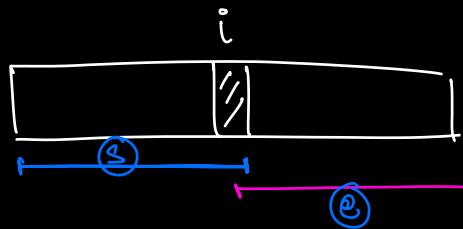
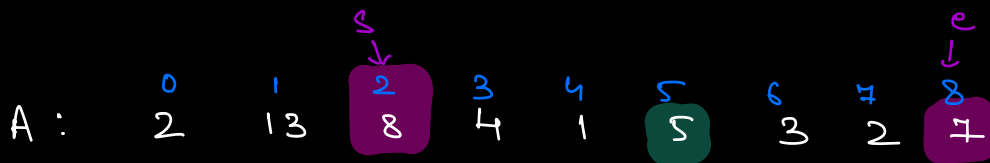
Contribution Technique.

For every element, find # of subarrays this element is present as MAX & MIN.

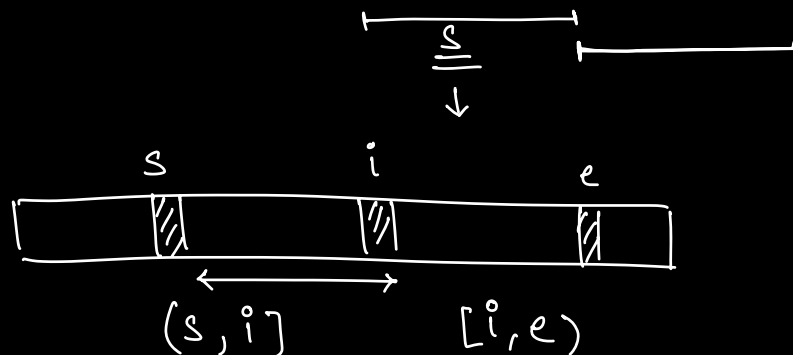
$A[i]$

* Find the count of subarrays in which $A[i]$ is present as Maximum.

* Find the count of subarrays in which $A[i]$ is present as minimum.



(s, e)



$[s, i]$
 \downarrow
 $i-s+1$

$\#$ of subarrays in which i will be present as MAX

$\Rightarrow (i-s) * (e-i)$

\uparrow \uparrow
NSL NSR

NSL of 5 $\leftarrow s$ $e \Rightarrow$ NSR of 5

A : 0 1 2 3 4 5 6 7 8
 2 13 8 4 1 5 3 2 7

$\#$ of subarrays in which i will be present as MIN

$\Rightarrow (i-s) * (e-i)$

\uparrow \uparrow
NSL NSR

$s = -1$ i $e = N$

A : 0 1 2 3 4 5 6 7 8
 2 13 8 4 1 5 3 2 7

$[s, i]$ $[i, e)$

$$(4 - (-1)) * (9 - 4)$$

$$5 * 5 = \underline{\underline{25}}$$

1. $NSL[] \leftarrow$ indices of Nearest smaller element on left side.

2. $NSR[]$

3. $NGL[]$

4. $NGR[]$

for ($i = 0; i < N; i++$) {

{
 $max = (i - \underline{NGL[i]}) * (\underline{NGR[i]} - i) * A[i]$
 $min = (i - \underline{NSL[i]}) * (\underline{NSR[i]} - i) * A[i]$
 $ans += (max - min);$

3
return ans;

TC : $O(N)$

SC : $O(N)$

— * —