Q Given an array. Find the nearest smaller element on left for every element. ↳ position wise (nearest index)

Ex1

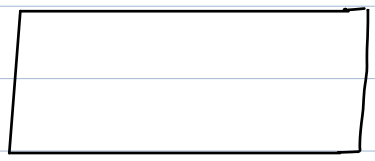
| | | | | | | |
|---|---|---|---|---|---|---|
| A[] : | 4 | 5 | 2 | 10 | 11 | 2 |
| NSE : | -1 | 4 | -1 | 2 | 10 | -1 |

Brute force : 2 loops : Tc: $O(n^2)$
Sc: $O(1)$

Ex2

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

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



Tc: $O(n)$
Sc: $O(n)$

Candidate list → Stack

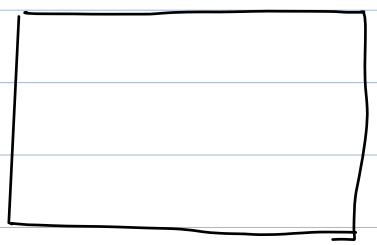
## Pseudo Code

```
int arr[n], int ans[n], stack<int> st.

for (int i=0; i<n ; i++) {.

    while (!st.empty() && st.top() ≥ arr[i])
        st.pop();

    if (st.empty())
        ans[i] = -1;
    else
        ans[i] = st.top();   ▽ returns the topmost element of stack.

    st.push(arr[i]);

}
```

Q2 Find the index of the NSL.

Ex1

| | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| A[] : | 2 | 7 | 6 | 3 | 1 | 5 |
| NSE : | -1 | 2 | 2 | 2 | -1 | 1 |
| | -1 | 0 | 0 | 0 | -1 | 4 |



| | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| A[] : | 2 | 7 | 6 | 3 | 1 | 5 |
| | -1 | 0 | 0 | 0 | -1 | 4 |

```
int arr[n], int ans[n], stack<int> st.

for (int i=0; i<n ; i++) {

    while (!st.empty() && arr[st.top()] >= arr[i])
        st.pop();

    if (st.empty())
        ans[i] = -1;
    else
        ans[i] = st.top();   // returns the topmost element of stack.

    st.push(i);

}
```

TC: O(n)

SC: O(n)

Q3 Given an array. For every element find the distance of it with it's NSL.

↳ Nearest smaller element

```
i - nsl[i]
```

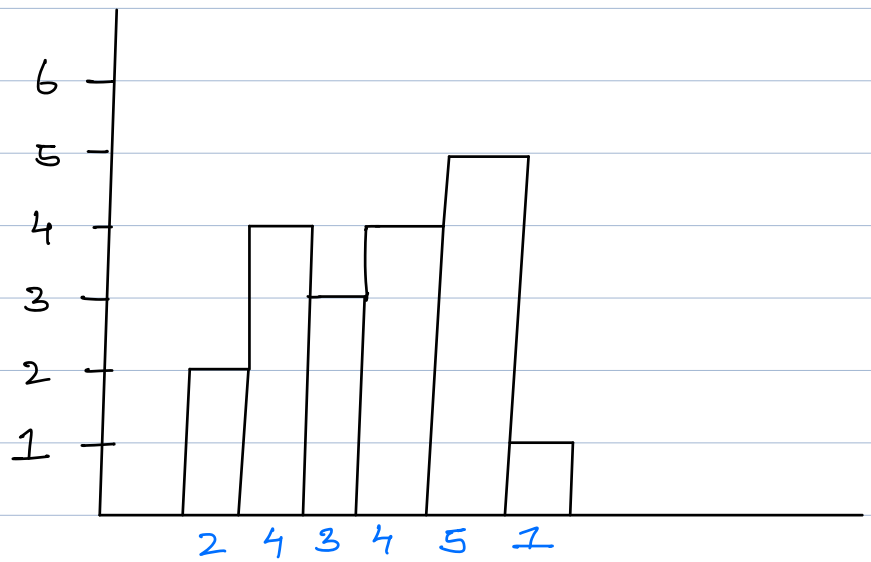
NSR = nearest smaller to right

Ex1

| | | | | | | |
|---|---|---|---|---|---|---|
| A[] : | 4 | 5 | 2 | 10 | 11 | 2 |
| nsR : | 2 | 2 | -1 | 2 | 2 | -1 |
| | 2 | 2 | -1 | 5 | 5 | -1 |

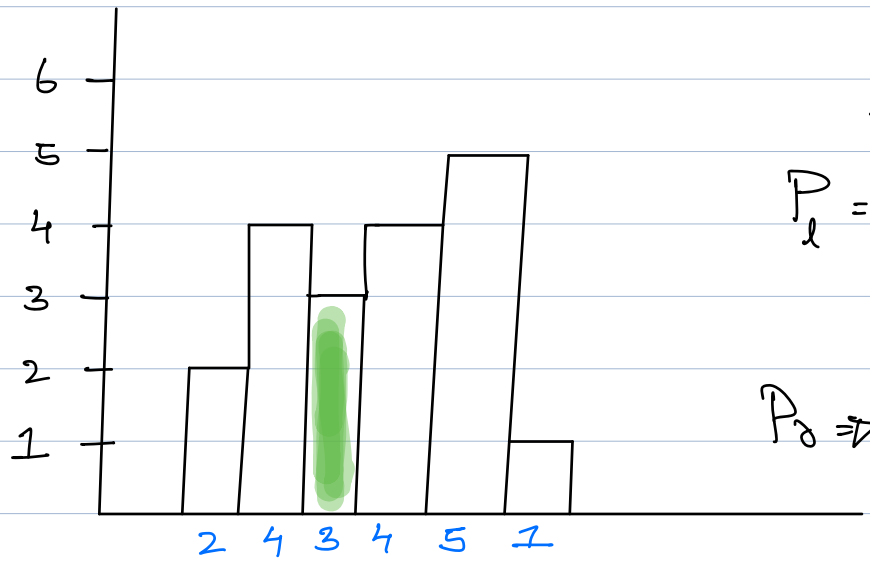
Q4 Given a histogram. Find the max rectangular area contained in it!

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




Brute Force : 1) Fix ends
2) Find min height b/w them
3) Find area

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




nsl +1

$P_l$ = index to which we expand on the left

$P_R$ ⇒ index to which we can expand on the right.

nsl −1 0 0 2 3 −1

nsR 5

if (nsl[i] == -1)
  $P_l$ ⇒ 0
else
  $P_l$ = nsl[i] + 1.

if (nsR[i] == -1)
  $P_R$ ⇒ n-1
else
  $P_R$ ⇒ nsR[i] - 1

# Fix every element as height & find $P_R$ & $P_L$.

Width ⇒ $(P_R - P_L + 1)$

height ⇒ arr[i];

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

Q5 Given a binary matrix. Find the max rectangular area with all 1's in it.

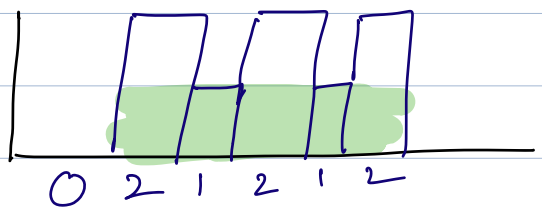
Ex1

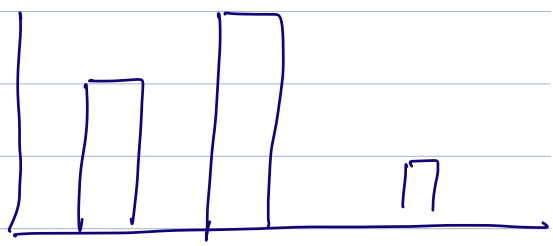
m

n

| | | | | | |
|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 |

| | | | | | |
|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 2 | 1 | 2 | 1 | 2 |
| 1 | 3 | 2 | 3 | 2 | 0 |
| 2 | 0 | 3 | 0 | 0 | 1 |
| 3 | 1 | 0 | 0 | 1 | 2 |




Tc: O(mn)

SC: O(m)

10:35

Q6 Similar to NSL, can we find NGL & NGR

NGL → Nearest greater to left

NGR → Nearest greater to right

| | | | | | | |
|---|---|---|---|---|---|---|
| A[] : | 4 | 5 | 2 | 10 | 11 | 2 |
| NGL : | -1 | -1 | 5 | -1 | -1 | 11 |
| NGR | 5 | 10 | 10 | 11 | -1 | -1 |

Q7 Given an array. Find the sum of max of every subarray!

Ex. arr[] = [1 4 3]

[1] = 1
[1 4] = 4
[1 4 3] = 4
[4 3] = 4
[4] = 4
[3] = 3

→ 20

Hint : Contribution technique.

| | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| | [ 10 | 2 | 7 | 6 | 11 | 9 ] |
| | ↑ | | | | ↑ | |
| | nGL | | | | nGR | |

if(nGL == -1) x ⇒ 0
if(nGR == -1) y ⇒ n-1

x ⇒ nGL + 1
y ⇒ nGR - 1

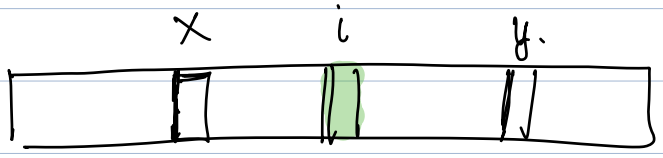
TC: O(n)

SC: O(n)

$$(i - x + 1) \times (y - i + 1)$$

$$(i - nGL + 1) \times (nGR - i + 1)$$

$$(i - nGL) \times (nGR - i)$$



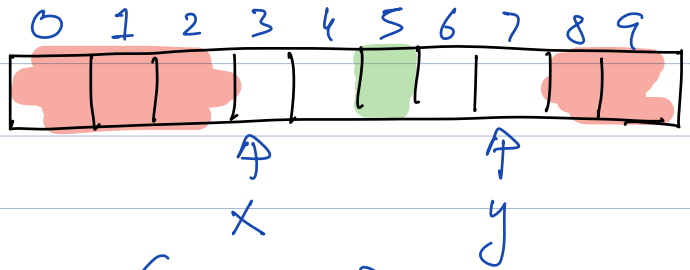

Starting points $\Rightarrow (i - x + 1)$

Ending points $\Rightarrow (y - i + 1)$

Total subarray which contain i

$\Rightarrow (i - x + 1) \times (y - i + 1)$

#




Starting $= (5 - 3 + 1) \Rightarrow 3 \rightarrow 3 \times 3 \Rightarrow 9$

Ending $= (7 - 5 + 1) \Rightarrow 3$