

“

The healthiest competition  
occurs when average people  
win by putting above average  
effort.

COLIN POWELL

GRACIOUSQUOTES.COM



Good  
Evening

### Today's content

01. Nearest smaller on left/right
02. Nearest greater on left/right
03. Largest Area histogram
04. Sum of (max-min) in all subarrays

01. Given an integer array A, find the index of nearest smaller element on left.

ar[6] = { 4 5 2 10 3 12 }

0 1 2 3 4 5

ans[6] = { -1 0 -1 2 2 4 }

ar[8] = { 4 6 10 11 7 8 3 5 }

0 1 2 3 4 5 6 7

ans[8] = { -1 0 1 2 1 4 -1 6 }

ar[] = { 8 2 4 9 7 5 3 10 }

0 1 2 3 4 5 6 7

ar[] = { -1 -1 1 2 2 2 1 6 }

Brute force → For all the elements at index i, iterate from i-1 to 0 & look for the first smaller ele

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

```
    ans[i] = -1;
```

```
    for (j=i-1; j>=0; j--) {
```

```
        if (A[j] < A[i]) {
```

```
            ans[i] = j;
```

```
            break;
```

TC :  $O(n^2)$

SC :  $O(1)$

```

    3
    |
    2
    |
return ans;

```

## \* Observation

$$A[] = \{ 8 \text{ --- } 5 \text{ } x \}$$

can idx 0 be the answer for  $x$ ?

- |                    |              |                |
|--------------------|--------------|----------------|
| 01. $x \leq 5$     | $ans \neq 0$ | } $ans \neq 0$ |
| 02. $5 < x \leq 8$ | $ans = 5$    |                |
| 03. $x > 8$        | $ans = 5$    |                |

$$ar[] = \{ 8 \quad 2 \quad 4 \quad 9 \quad 7 \quad 5 \quad 3 \quad 10 \}$$

0    1    2    3    4    5    6    7

$$ans[] = -1 \quad -1 \quad 1 \quad 2 \quad 2 \quad 2 \quad 1 \quad 6$$

for a particular ele,

we need to look for the latest ele on left.

<del>8</del>	2	<del>4</del>	<del>9</del>	<del>7</del>	<del>5</del>	3	10
--------------	---	--------------	--------------	--------------	--------------	---	----

↓  
 $ar[6]$

(i)

```
for (i=0 ; i<n; i++)
```

```
    while (st.size() > 0 && A[st.peek()] ≥ A[i])
```

```
        | st.pop();
```

```
    if (st.size() == 0) ans[i] = -1
```

```
    else    ans[i] = st.peek();
```

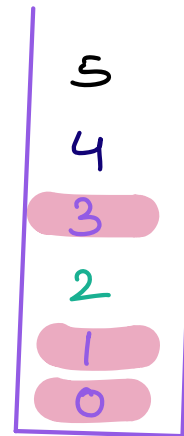
```
    st.push(i);
```

TC:  $O(n)$

SC:  $O(n)$

arr = { 4 5 2 10 3 12 }

ans = { -1 0 -1 2 2 4 }



02.  $\forall i$ , find nearest smaller or equal on left

$A[st.peek()] > A[i]$

03.  $\forall i$ , find nearest greater ele on left

$A[st.peek()] \leq A[i]$

04.  $\forall i$ , find nearest greater or equal ele on left

$A[st.peek()] < A[i]$

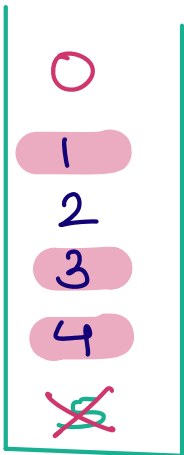
Q  $\forall i$ , find nearest smaller on the right.

```
for ( i = n-1 ; i >= 0 ; i-- )  
    while ( st.size() > 0 && A[st.peek()] >= A[i] )  
        st.pop();  
    if ( st.size() == 0 ) ans[i] = -1  
    else ans[i] = st.peek();  
    st.push(i);
```

ans[6] = { 2 2 -1 4 -1 -1 }

↓  
arr[6] = { 4 5 2 10 3 12 }

0 1 2 3 4 5



\* 3 variation  $\rightarrow$  Do it by yourself

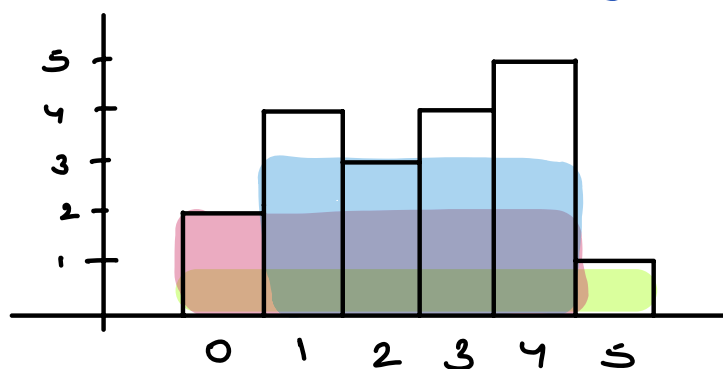
9:54 pm → 10:04 pm

## Histogram Area

Given continuous block of Histogram find max Rectangular area which can be present within histogram

Note:- Every histogram is of width = 1

Ex :  $ar[6] = \{2, 4, 3, 4, 5, 1\}$



Areas

6

10

12

✓ Ans

Brute force → ~~all~~ all subarrays as base, find

max height & calculate area

min ele in  
subarray

TC =  $O(n^3)$

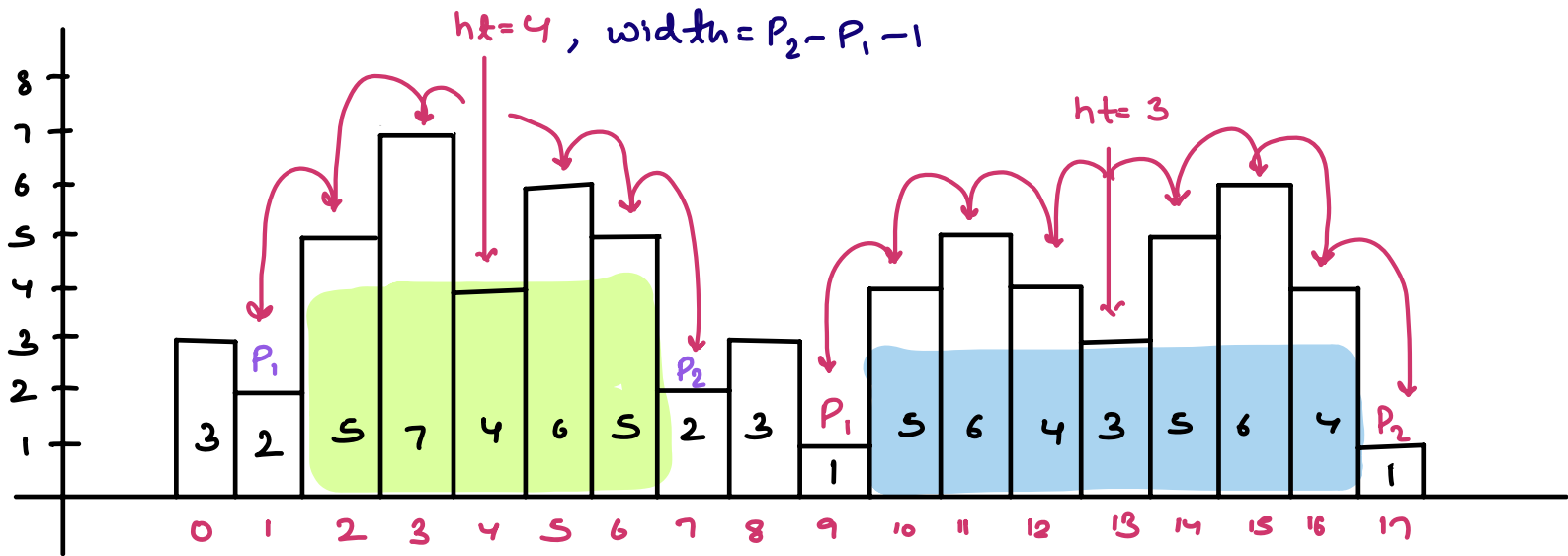
SC =  $O(1)$

↓ Carry forward

$O(n^2)$

arr[] = 

3	2	5	7	4	6	5	2	3	1	5	6	4	3	5	6	4	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---



$$\begin{aligned} \text{area} &= 4 * 5 \\ &= 20 \end{aligned}$$

$$\begin{aligned} \text{area} &= 3 * 7 \\ &= 21 \checkmark \end{aligned}$$

Observation → Height of the rectangle is always going to be height of one of the histogram.

Idea → Iterate on each & every histogram, considering the histogram as the height of rectangle

- ① → Iterate on left & find first smaller element
- ② → Iterate on right & find first smaller element

$$\begin{aligned} \text{Area} &= \text{ht} * (P_2 - P_1 - 1) \\ &\rightarrow \text{Maximise the area.} \end{aligned}$$

```
int RectangleArea (int [] ar)
```

```
int [] left = nearest smaller ele on left (ar);
```

```
int [] right = nearest smaller ele on right (ar);
```

```
area = 0
```

→ initialise ans = 0

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

```
    h = ar[i]
```

```
    P1 = left[i]
```

```
    P2 = right[i]
```

```
    width = P2 - P1 - 1;
```

```
    area = Max (area, h * width);
```

```
}
```

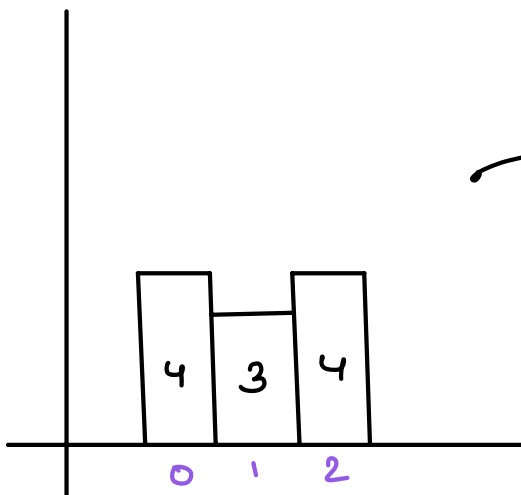
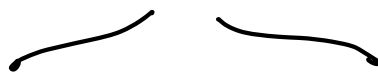
```
return ans
```

```
}
```

TC:  $O(n)$

SC:  $O(n)$

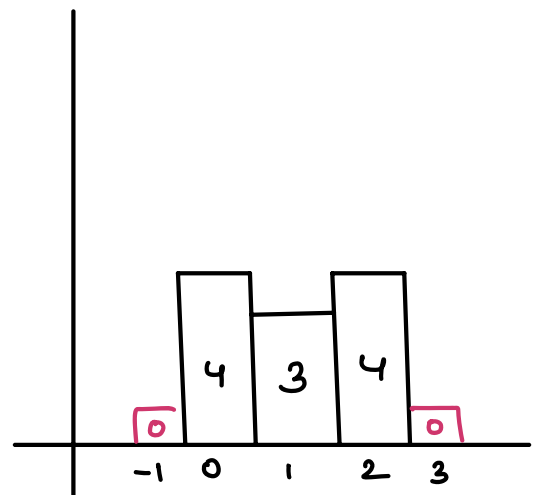
Edge



$P_1 = -1 \quad -1 \quad 1$

$P_2 = 1 \quad -1 \quad -1$

width = 1    -1    -3



$P_1 = -1 \quad -1 \quad 1$

$P_2 = 1 \quad 3 \quad 3$

width = 1    3    1



$$P_2 - P_1 - 1 \rightarrow -1 - 1 - 1 \\ = -3$$

$$\begin{array}{ll} \downarrow & 3 - 1 - 1 = 1 \\ 1 - (-1) - 1 & 3 - (-1) - 1 \\ 2 - 1 & 4 - 1 \\ \rightarrow 1 & = 3 \end{array}$$

Q Given an integer array with distinct integers, find the sum of (max-min) for all subarrays as the answer.

$$A = \begin{Bmatrix} 2 & 5 & 3 \\ 0 & 1 & 2 \end{Bmatrix}$$

subarrays

$$2 \xrightarrow{\text{max} \quad \text{min}} 2 - 2 = 0$$

$$2, 5 \rightarrow 5 - 2 = 3$$

$$2, 5, 3 \rightarrow 5 - 2 = 3$$

$$5 \rightarrow 5 - 5 = 0$$

$$5, 3 \rightarrow 5 - 3 = 2$$

$$3 \rightarrow 3 - 3 = 0$$

$$\underline{\underline{\text{Ans} = 8}}$$

Brute force

for all subarrays,

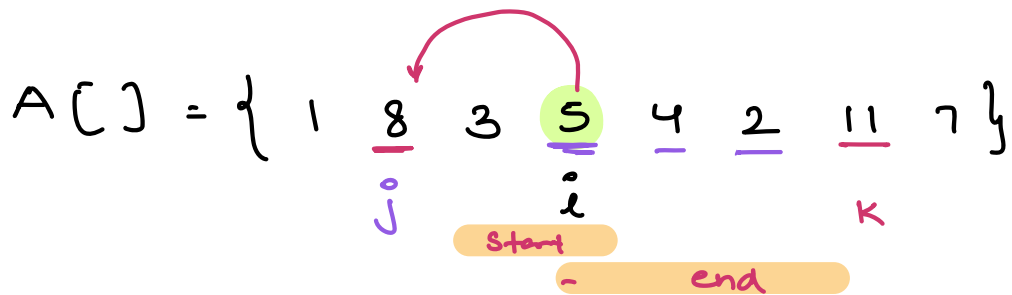
calculate  $\sum (\text{max} - \text{min})$

$$TC = O(n^2) \quad SC = O(1)$$

\* Contribution technique

$$\text{Ans} = \sum_{i=0}^{n-1} \text{contribution of } A[i]$$

$$A[i] * \left( \begin{array}{cc} \# \text{ of subarrays} & - \# \text{ of subarrays} \\ \text{where } A[i] \text{ is} & \text{where } A[i] \text{ is} \\ \text{maximum} & \text{minimum} \end{array} \right)$$



3 5                      5  
 3 5 4                  5 4  
 3 5 4 2                5 4 2

$j$  = nearest greater ele on left side of  $i$

$k$  = nearest greater ele on right side of  $i$

$$\text{starts} = [j+1 \quad i] \Rightarrow i - j$$

$$\text{end} = [i \quad k-1] = k - i$$

$$\# \text{ subarrays} = (i - j) * (k - i)$$

where  $A[i]$  is max

$$\text{Ans} = \sum_{i=0}^{n-1} A[i] * \left( (i - \text{great left}[i]) * (\text{greater right}[i] - i) - (i - \text{smaller left}[i]) * (\text{smaller right}[i] - i) \right)$$

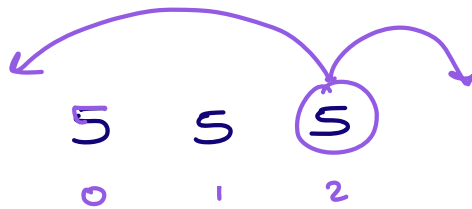
Tc:  $O(n)$

Sc:  $O(n)$

# of subarrays  
where  $A[i]$  is minimum

\* End of the class.  $\rightarrow$  Thankyou 😊

Doubt Session



0  
0-1  
0-1-2

0-1  
0-1-2  
1  
1-2

0-1-2  
1-2  
2

\* Infix to postfix  $\rightarrow$  Thursday optional