

# Stack Problems

[Nearest Smaller / Nearest Greater](#)

[Largest Rectangle in Histogram](#)

[Sum of \(max-min\) of all subarrays](#)

# Nearest Smaller / Nearest Greater

Array of size  $N$ . For every index  $i$ , find the nearest smaller element which is smaller than  $i$ th element on left side.

$arr_1 : \begin{matrix} 4 & 5 & 2 & 10 & 3 & 2 \\ x & 4 & x & 2 & 2 & x \end{matrix}$   
 $arr_2 : \begin{matrix} 4 & 6 & 10 & 11 & 7 & 8 & 3 & 5 \\ -1 & 4 & 6 & 10 & 6 & 7 & -1 & 3 \end{matrix}$

Brute force :-

- Iterate & find answer for every index.

```

for (i = 0; i < N; i++)
    for (j = i; j >= 0; j--)
        if (arr[i] > arr[j])
            return arr[j];
    
```

T.C:  $O(N^2)$

Stack Approach

$arr : \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 5 & 2 & 8 & 10 & 6 & 1 \\ -1 & -1 & 2 & 8 & 2 & -1 \end{matrix}$

5 2 8 10 6 1

Use Stack

→ access elements from this end  
 → delete element from this end.

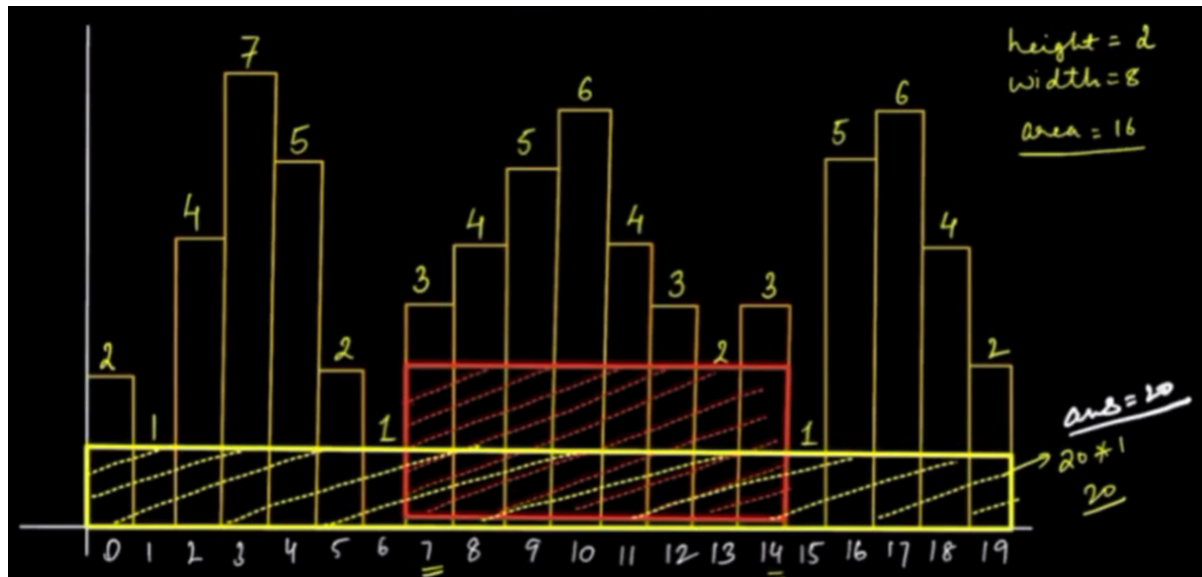
Code :-

```

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

Check for all different variants of this problem.

# Largest Rectangle in Histogram



1) Brute Force : Consider all possible points & find area.

```

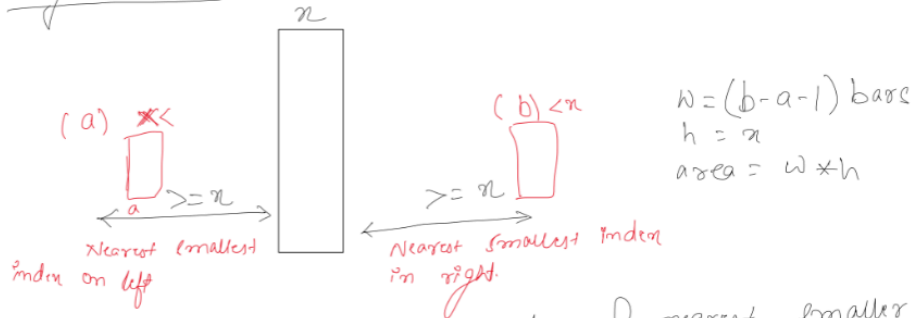
for (i=0; i<N; i++)
{
    for (j=i; j<N; j++) {
        // [i to j]
        h = min value from i to j.
        w = j - i + 1
        ans = max(ans, w * h);
    }
}

```

T.C:  $O(N^3)$

2) Using Carry forward  
 $h = \min(h, a[i]);$

3) Optimized solution



1) Create an array to store index of nearest smaller element on left.

2) Create an array to store index of nearest smaller element on right.

```

for (int i=0; i<N; i++) {
    int h = arr[i];
    int w = NsR[i] - NsL[i] - 1
    ans = max(ans, w * h)
}
return ans;

```

T.C:  $O(N)$   
 S.C:  $O(N)$

# Sum of (max-min) of all subarrays

Q Find sum of (max-min) for all subarrays.

|          |     |     |          |  |
|----------|-----|-----|----------|--|
|          | 0   | 1   | 2        |  |
| arr[3] : | 2   | 5   | 3        |  |
|          | 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 = ans. |  |

Brute Force

Consider all subarrays calculate (max-min) using carry forward ( $O(n^2)$ )

Approach 2

$$\textcircled{1} \sum \text{max}_s - \sum \text{min}_s = \sum (\text{max} - \text{min})$$

$\textcircled{2}$  using Contribution technique, calculate contributions.

$$\sum \text{max}_s = \sum_{i=0}^{N-1} \text{In how many subarrays } i^{\text{th}} \text{ element is max} \times \text{arr}[i]$$



Nearest greater on left:  $[NGL]$   $i^{\text{th}}$   $\text{Nearest greater on right: } [NGR]$   
Valid starts:  $[l+1, i]$   $\text{Valid ends: } [i, r-1]$

$$\# \text{ of subarrays} = \left( \frac{(i-1)}{1} \right) \times \left( \frac{(r-i)}{1} \right)$$

$$\sum \text{max}_s = \sum_{i=0}^{N-1} \left( (i - NGL[i]) \times (NGR[i] - i) \right) \times \text{arr}[i];$$

arr[] : { 2 5 3 }  
NGL[] : { -1 -1 1 }  
NGR[] : { 1 3 3 }

Simplify

$$\sum \text{min}_s = \sum_{i=0}^{N-1} \left( (i - NGL[i]) \times (NGR[i] - i) \right) \times \text{arr}[i]$$

Iterate & find the answer!