

# • Stacks 1



## ~~~~~ Agenda ~~~~~

1. Revision Quiz
2. Nearest Smaller Element on left 
3. Nearest greater [left + right]
4. Scenerio Based Problem : Restraunt Hunt
5. Largest Rectangle in Histogram
6. Sum of Max-Min in all subarray



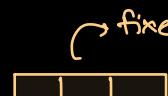
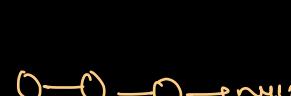
Hello Everyone  
Very Special Good Evening  
to all of you 😊😊😊  
We will start discussion  
from 09:06 PM

## ~~~~~ Revision Quiz ~~~~~

Quiz 1 :

What is the time complexity for push and pop operations in a stack?

- O(1)       O(log(N))  
 O(N)       O(N^2)



Quiz 2 :

What is the main advantage of using a linked list for stack implementation over an array?

- Faster access time  
 Dynamic size  
 Less memory usage  
 Simplicity

Quiz 3 :

In a postfix expression evaluation, which data structure is commonly used?

- Queue  
 Stack  
 Graph  
 Tree

#### Quiz 4 :

Which of the following is NOT a valid operation on a stack?

- [ ] Push
- [ ] Pop
- [ ] Peek
- [ ] Enqueue

#### Nearest Smaller Element on left OR next smaller

Given an integer array A, find the index of nearest smallest element on left for all i index in A[].

Formally , for all i find j such that  $A[j] < A[i]$ ,  $j < i$  and j is maximum.

$A: [ \begin{matrix} 10 & 16 & 5 & 9 & 12 & 8 & 25 & 7 & 13 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{matrix} ]$

next smaller (index)	-1	0	-1	2	3	2	2	2	7
	-1	10	-1	5	9	5	8	5	7

$A: [ \begin{matrix} 18 & 3 & 13 & 19 & 5 & 24 & 4 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} ]$

next smaller (index)	-1	-1	1	2	1	4	1
----------------------------	----	----	---	---	---	---	---

#### Quiz 1:

Given N array elements, find the index of the nearest smaller element on the left side for all the elements. If there is NO smaller element on left side, ans is -1. A = [4, 6, 10, 11, 7, 8, 3, 5]

arr: [4, 6, 10, 11, 7, 8, 3, 5]  
 $\begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix}$

nsl  $\rightarrow$  -1 0 1 2 1 4 -1 6  
 (index)  
 $\begin{matrix} & & & & & & \uparrow \\ & & & & & & i \end{matrix}$

#### Quiz 2:

$A = [4, 5, 2, 10, 8, 2]$   
 $\begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 \end{matrix}$   
 nsl  $\rightarrow$  -1 0 1 2 2 1  
 (index)  $\begin{matrix} & & & & & \uparrow \\ & & & & & i \end{matrix}$

Bruteforce (next smaller on left - index Based):

```

result[n];
for(int i=0; i<n; i++) {
    nextSmallerIndex = -1;
    for(int j= i-1; j>=0; j--) {
        if( arr[j] < arr[i] ) {
            nextSmallerIndex = j;
            break;
        }
    }
    result[i] = nextSmallerIndex;
}

```

arr: [4, 6, 10, 11, 7, 8, 3, 5]  
 $\begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \downarrow & \downarrow \\ -1 & 0 & 1 & 2 & 1 & 4 & -1 & 6 \end{matrix}$

T.C:  $O(n^2)$   
S.C:  $O(1)$

Quiz!

If A = [8, x, x, x, x, 5, x, x, x, x...]

For any element present after 5, can the element 8 become nearest smaller element on left?

No, because 8 is available on right which is already smaller than 8.

Optimise Approach: [next smaller element on left side - index]

A: [  $\frac{\cancel{10}}{0}$   $\frac{\cancel{16}}{1}$   $\frac{\cancel{5}}{-1}$   $\frac{\cancel{9}}{2}$   $\frac{\cancel{12}}{3}$   $\frac{\cancel{8}}{4}$   $\frac{\cancel{25}}{5}$   $\frac{\cancel{7}}{6}$   $\frac{\cancel{13}}{7}$  ]

Iteration count  $\rightarrow 2n$   
(c worst case)

T.C:  $O(n)$

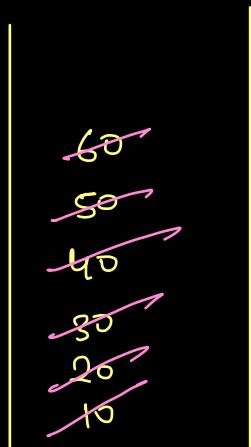
S.C:  $O(n)$

$\frac{13-8}{7-7}$   
 $\cancel{\frac{25-6}{8-5}}$   
 $\cancel{\frac{12-4}{9-3}}$   
 $\cancel{\frac{5-2}{16-1}}$   
 $\cancel{\frac{10-0}{10-0}}$

$$A: \begin{bmatrix} \checkmark & \checkmark & \checkmark & \checkmark & \checkmark & \checkmark & \checkmark \\ 18 & 3 & 13 & 19 & 5 & 24 & 4 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline -1 & -1 & \perp & 2 & \perp & 4 & \perp \end{bmatrix}$$

$$\text{arr}: \begin{bmatrix} \checkmark & \checkmark & \checkmark & \checkmark & \checkmark & \checkmark & \boxed{2} \\ 10 & 20 & 30 & 40 & 50 & 60 & 6 \\ 0 & 1 & 2 & 3 & 4 & 5 & \boxed{6} \\ \hline -1 & \perp & \perp & \perp & \perp & \perp & \perp \end{bmatrix}$$

$$\begin{array}{r} 4-6 \\ \hline 24-5 \\ \hline 5-4 \\ \hline 19-3 \\ \hline 13-2 \\ \hline 3-1 \\ \hline 18-0 \end{array}$$



$$\text{Inv Count} = n + n \\ = 2n$$

$T.C: O(n)$

$S.C: O(n)$

$$\begin{bmatrix} 30 & 20 & 10 \\ \hline -1 & -1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 10 \\ \hline 20 \\ \hline 30 \end{bmatrix}$$

Steps:

- ① How many Element is there in the Stack which are greater than arr[i]  $\rightarrow$  pop them.
- ② If peek of stack is less than arr[i], then peek value is next smaller, (peek index is next smaller index)
- ③ while we are removing count & not yet available in stack then next smaller will be -1
- ④ push arr[i] or  $\begin{smallmatrix} i \\ \text{val based} \end{smallmatrix}$  or  $\begin{smallmatrix} i \\ \text{index based} \end{smallmatrix}$  in the stack

```

int[] nextSmallerLeftIndex(int[] arr, int n) {
    int[] ans = new int[n];
    Stack<Integer> st = new Stack<>();
    st.push(0);
    ans[0] = -1;
    for (int i = 1; i < n; i++) {
        while (st.size() > 0 && arr[st.peek()] >= arr[i]) {
            st.pop();
        }
        if (st.size() == 0) {
            ans[i] = -1;
        } else {
            ans[i] = st.peek();
        }
        st.push(i);
    }
    return ans;
}

```

T.C: O(n)  
S.C: O(h)

next Smaller on Right (index + elements based)

A: [ 10 16 5 9 12 8 25 7 3 ]

↳ exactly same algo → Start from Right end.

$\left\{ \begin{array}{l} \text{next greater on left} \\ \text{next greater on Right} \end{array} \right\}$  → change the condition in  
 algo of next smaller  $\left\{ \begin{array}{l} \text{left} \\ \text{Right} \end{array} \right\}$

TODD

Scenerio Based Problem : Restraunt Hunt

A person uses Google Maps to find the nearest restaurants and picks one based on its proximity. Unfortunately, after visiting, they realized that the restaurant didn't meet their expectations.

## Task

Let's break it down with a simple example. You have a list of restaurants and their ratings. For each restaurant, we're going to find the next restaurant to the right on the list that's not just close but also has a higher rating than the current one. If there's no better option on the list, we'll say there's none available.

## Problem

Given a sequence of restaurants listed on Google Maps with their ratings, create a tool that helps users discover the rating of the next higher-rated restaurant to the right for each listed establishment.

# Solve it for next greater on Right (TO DO)

## Largest Rectangle in Histogram

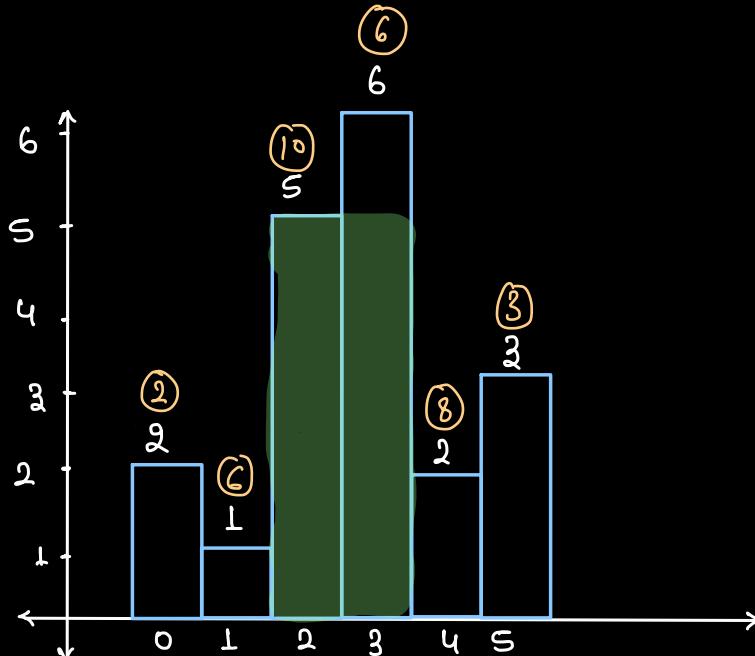
Given an integer array A, where

$A[i]$  = height of i-th bar.

Width of each bar is = 1.

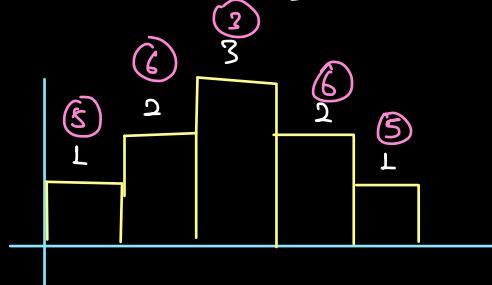
Find the area of the largest rectangle formed by continuous bars.

$$A: [ \begin{matrix} 2 & 1 & 5 & 6 & 2 & 3 \end{matrix} ]$$

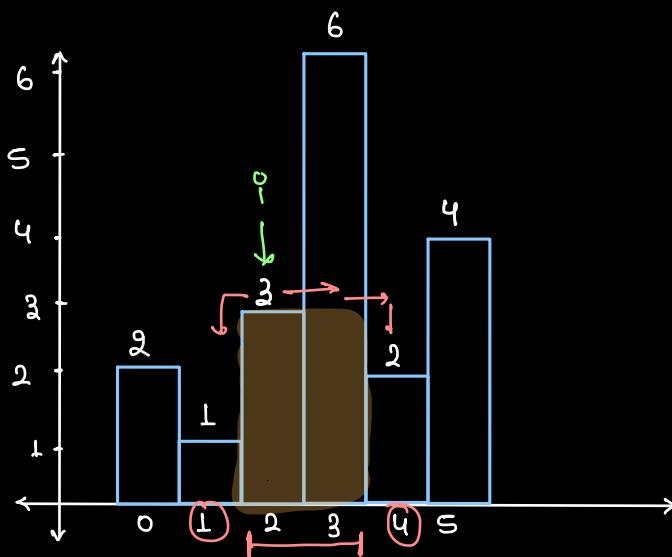


largest Rect = 10 sq. unit

$$\text{Quiz: } [1, 2, 3, 2, 1]$$



max area: 6 sq. unit



$$A' [ \begin{matrix} 2 & 1 & 3 & 6 & 2 & 4 \end{matrix} ]$$

for  $i=2$

width = ? ②

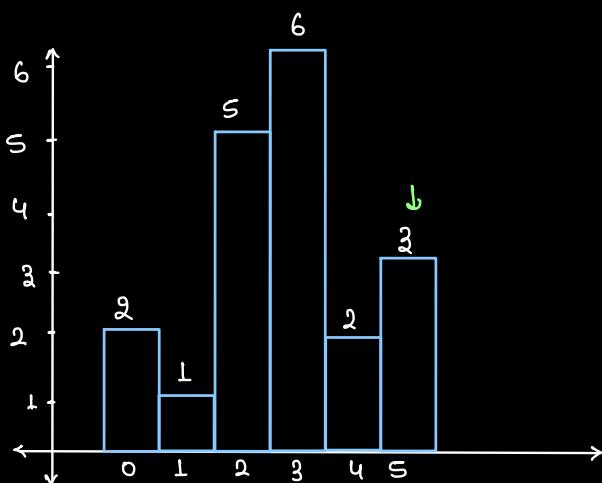
left smaller element index = 1

right smaller element index = 4

width =  $4 - 1 - 1 = ②$

height = arr[i] : ③, Area = h \* w =  $3 \times 2 = 6$  sq.u

$$A: \left[ \begin{smallmatrix} 2 & 1 & 5 & 6 & 2 & 3 \\ 0 & 1 & 2 & 3 & 4 & 5 \end{smallmatrix} \right]$$

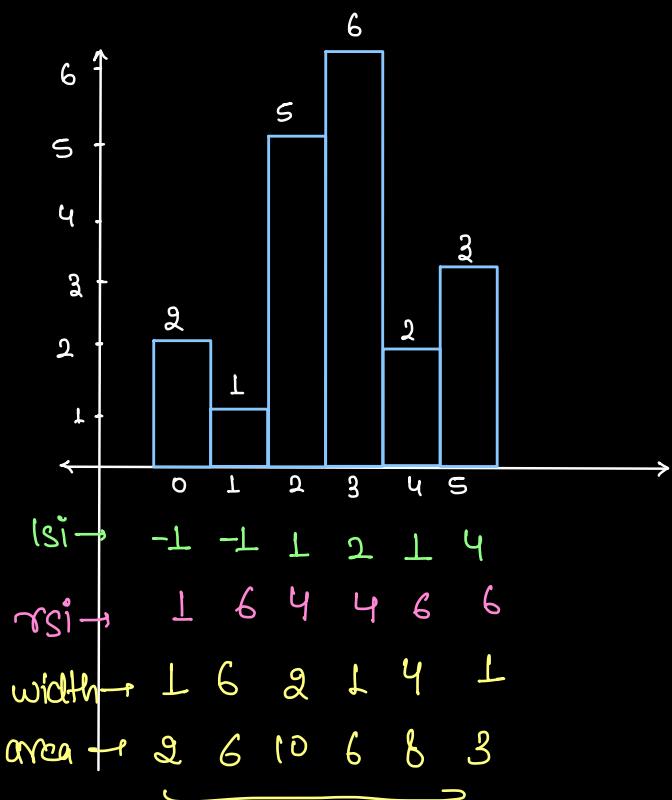


i	ht	lsi	rsi	width	area
0	2	-1	1	1	2 sq. unit
1	1	-1	6	6	6 sq. unit
2	5	1	4	2	10 sq. unit
3	6	2	4	1	6 sq. unit
4	2	1	6	4	8 sq. unit
5	3	4	6	1	3 sq. unit

(6) → loop stop

ans = max = 10 sq. unit

$$\text{arr}: [2, 1, 5, 6, 2, 3]$$



pseudo code:

```

int() lsi = nextSmallerOnLeft(arr);
int() rsi = nextSmallerOnRight(arr);

ans = 0;
for(int i=0; i<n; i++) {
    ht = arr[i];
    width = rsi[i] - lsi[i] - 1;
    area = ht * width;
    ans = max(ans, area);
}
    
```

return ans;

Break  
10:48 - 10:58 pm

## Sum of Max-Min in all subarray

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

arr: [2, 5, 3, 4]

all Sub arrays:

	max	min	diff	
2	2	2	0	
2 5	5	2	3 ↗	sum of diff(max - min)
2 5 3	5	2	3 ↗	= 3 + 3 + 3 + 2 + 2 + 1
2 5 3 4	5	2	3 ↗	= 14 ↗
5	5	5	0	
5 3	5	3	2 ↗	
5 3 4	5	3	2 ↗	
3	3	3	0	
3 4	4	3	1 ↗	
4	4	4	0	

Bruteforce:

0 1 2

i=0 [0] [1] [2]

carry  
forward  
Approach

Sum = 0;

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

```
    for(int j=i; j<n; j++) {
        // 'i' is start index & 'j' is end index of subarray
        // find max & min b/w i to j
        maxEle = arr[i];
        minEle = arr[i];
```

```
        for(int k=i; k<=j; k++) {
            maxEle = max(maxEle, arr[k]);
```

```
            minEle = min(minEle, arr[k]);
        }
```

```
        sum += (maxEle - minEle);
```

TC: O(n<sup>3</sup>)  
SC: O(1)

return sum

Ques:

[1, 2, 3]

Sum of  $\text{diff}(\max - \min)$  ?

Subarray of length = 1

$$[1] \rightarrow \max = 1, \min = 1 \rightarrow \text{diff} = 0$$

$$[2] \rightarrow \max = 2, \min = 2 \rightarrow \text{diff} = 0$$

$$[3] \rightarrow \max = 3, \min = 3 \rightarrow \text{diff} = 0$$

Subarray of length = 2

$$[1, 2] \rightarrow \max = 2, \min = 1 \rightarrow \text{diff} = 1$$

$$[2, 3] \rightarrow \max = 3, \min = 2 \rightarrow \text{diff} = 1$$

Subarray of length = 3

$$[1, 2, 3] \rightarrow \max = 3, \min = 1 \rightarrow \text{diff} = 2$$

$$\text{Sum} = 0 + 1 + 1 + 2 = 4 \quad \underline{\underline{\text{ans}}}$$

## Optimised Solution!

Ans: [1, 2, 3]

$\max - \min$

$$1 \text{ } \underline{\underline{\underline{\quad}} \quad} 1-1 = 0$$

ans = 4

$$1 \text{ } 2 \text{ } \underline{\underline{\underline{\quad}} \quad} 2-1 = 1$$

$$1 \text{ } 2 \text{ } 3 \text{ } \underline{\underline{\underline{\quad}} \quad} 3-1 = 2$$

$$2 \text{ } \underline{\underline{\underline{\quad}} \quad} 2-2 = 0$$

$$2 \text{ } 3 \text{ } \underline{\underline{\underline{\quad}} \quad} 3-2 = 1$$

$$3 \text{ } \underline{\underline{\underline{\quad}} \quad} 3-3 = 0$$

crr $\rightarrow [L$	2	$3]$
freq. of arr[i] ① as max in Subarray =	②	③
freq. of arr[i] = ③ as min in Subarray	②	①
contribution in sum $\rightarrow 1*1 - 3*1$	$2*2 - 2*2$	$3*3 - 3*1$
$\Rightarrow \textcircled{-2} + \textcircled{0} + \textcircled{6}$		$\Rightarrow \textcircled{4} \cancel{\textcircled{14}}$

B was greater from index till item B [a, b, c, d] B is max until we will not find next greater than B in Right

it was min c is min until next element is min

till next smaller in left is not there.

for any particular value of arr[i], required info:

- \* next greater on left
  - \* next greater on right
  - \* next smaller on left
  - \* next smaller on right
- $\left. \begin{matrix} \\ \\ \\ \end{matrix} \right\} \text{TODD} \rightarrow \text{get these info}$

freq. in which b is treated as max = ?  
freq. in which b is treated as min = 0

arr:	[ 2, 0,	$\frac{5}{2},$	3, 2	4 ] 3
prev greater $\rightarrow$	-1	-1	1	1
next greater $\rightarrow$	1	4	3	4
prev smaller $\rightarrow$	-1	0	0	2
next smaller $\rightarrow$	4	2	4	4

$$\text{Contribution of } arr[i] \text{ as min} = (next \text{ smaller} - i) * (i - prev \text{ smaller})$$

Similarly,

$$\text{Contribution of } arr[i] \text{ as max} = (next \text{ greater} - i) * (i - prev \text{ greater})$$

$$\text{Total contri} = [contri \text{ as max} - contri \text{ as min}] + arr[i]$$

arr:	[ 2, 0,	$\frac{5}{2},$	3, 2	4 ] 3
prev greater $\rightarrow$	-1	-1	1	1
next greater $\rightarrow$	1	4	3	4
prev smaller $\rightarrow$	-1	0	0	2
next smaller $\rightarrow$	4	2	4	4

i	contri as min	contri as max	total contri
0 $\rightarrow$	$(4-0)* (0-(-1))$ $4*1 = 4$	$(1-0)* (0-(-1))$ $1*1 = 1$	$2*(1-4) = -6$
1 $\rightarrow$	$(2-1)* (1-0)$ $1*1 = 1$	$(4-1)* (1-(-1))$ $3*2 = 6$	$5*(5-1) = 20$
2 $\rightarrow$	$(4-2)* (2-0)$ $2*2 = 4$	$(3-2)* (2-1)$ $1*1 = 1$	$3*(1-4) = -9$
3 $\rightarrow$	$(4-3)* (3-2)$ $1*1 = 1$	$(4-3)* (3-1)$ $1*2 = 2$	$4*(2-1) = 4$

$$\text{Sum} = 14$$

We have problem solving session on Sunday @ 10:00 AM

Agenda: (optional) → Reading available.

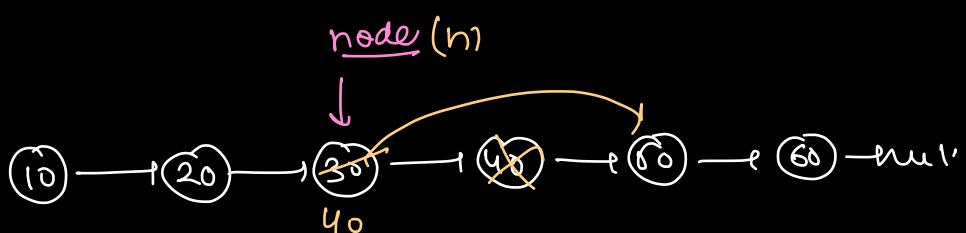
2 hr

\* Clone of LL

\* Medium of array

\* Implementation of LL

→ normal    { implement in JAVA  
     OOPS }



Set the data of curr node same as next node

$n.data = n.next.data;$

delete next node

$n.next = n.next.next;$

Recursion: Dry Run

Recursion: Represent  
{ Xanthu }

please explain this syntax sir for java  
 $v[s.charAt(i) - 'a'] += (\text{adjSign}(s, i) ? \boxed{\text{add} ? -1 : 1} : \boxed{\text{add} ? 1 : -1});$

ternary opr.

expression ? Statement : Statement;