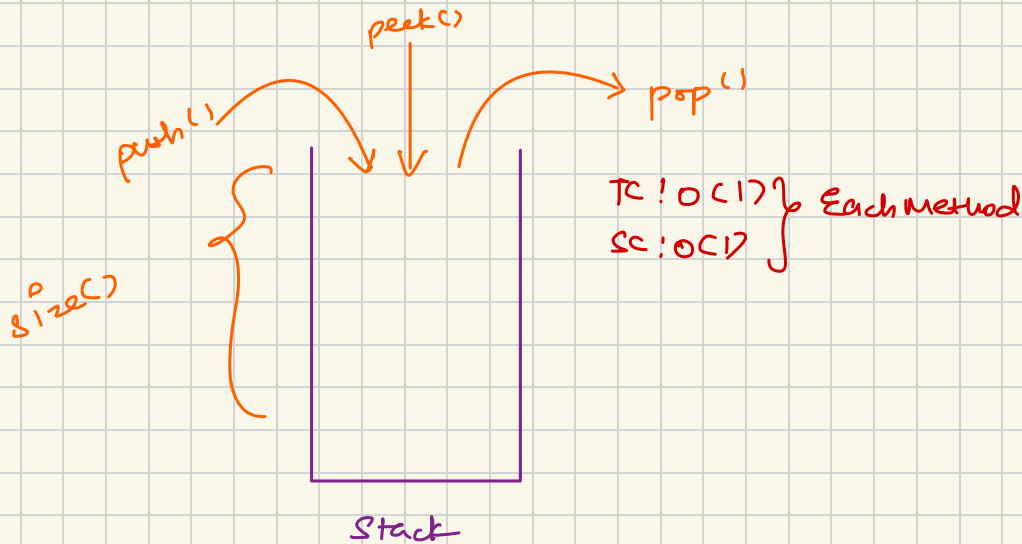




# Stacks

- Linear data structure
- LIFO { Last In, first out }



## Agenda:

- ① Extra Brackets
- ② Next greater Ele on Right
- ③ Stock span
- ④ Largest Area Histogram

## Extra Brackets .

string exp = "(a+b)" No Extra Bracket

= "((a+b))" YES Extra Bracket

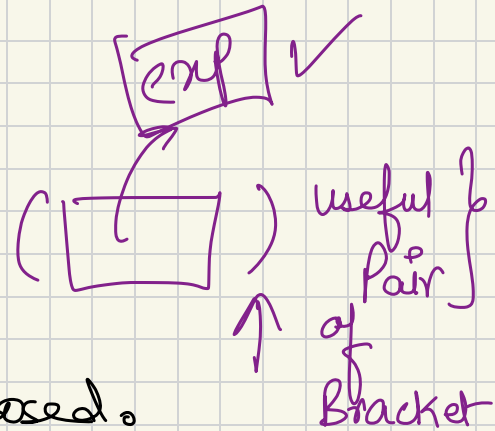
NOTE: A Bracket pair is useful if it has a new expression enclosed in it.

= "(a+b)\*(d+e-(f/g)\*(h))" No Extra Brackets

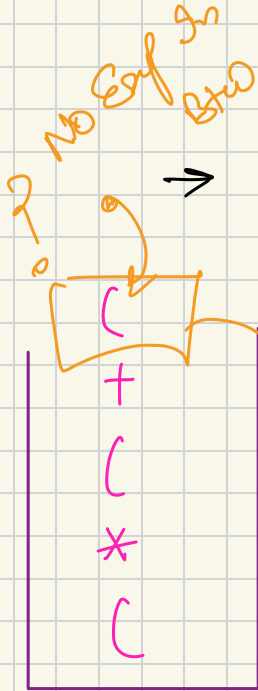
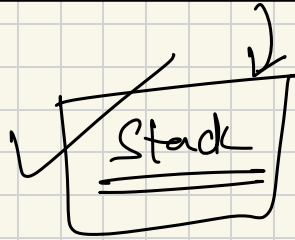
= "((a)+(b))" No Extra Bracket

$$\text{exp} = \left( (a+b) * (d) + ((f+g)) \right)^n$$

↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑



→ last opened bracket, first to be closed.

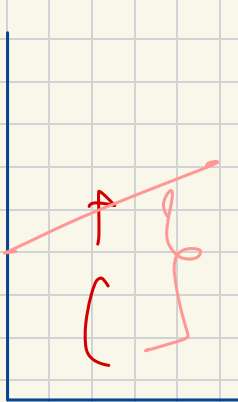


→ found an Extra Bracket Pair

$$\text{exp} = {}^u \left( (a) + (b) \right)^u$$

$\uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow$

$\boxed{\text{Exp}} \rightarrow \text{in b/w}$



stack

✓ No Extra Bracket here!

## Next greater Element On Right

int[] arr = { 3, 6, 1, 2, 7, 4, 5 }

✓ nger[] = { 6, 7, 2, 7, -1, 5, -1 }

o/p

- find ele on right nearest and greater to the current Element -

Brute force

Nested loop. TC:  $O(N^2)$  SC:  $O(1)$

arr = {<sup>0</sup>3, <sup>1</sup>6, <sup>2</sup>1, <sup>3</sup>2, <sup>4</sup>7, <sup>5</sup>3, <sup>6</sup>4, <sup>7</sup>1, <sup>8</sup>2, <sup>9</sup>6}

↑    ✗   ✗   ✗   ✗   ✗   ✗   ✗   ✗   ✗

{ 6   7   2   7   -1   4   6   2   6   -1 }

3  
6  
7

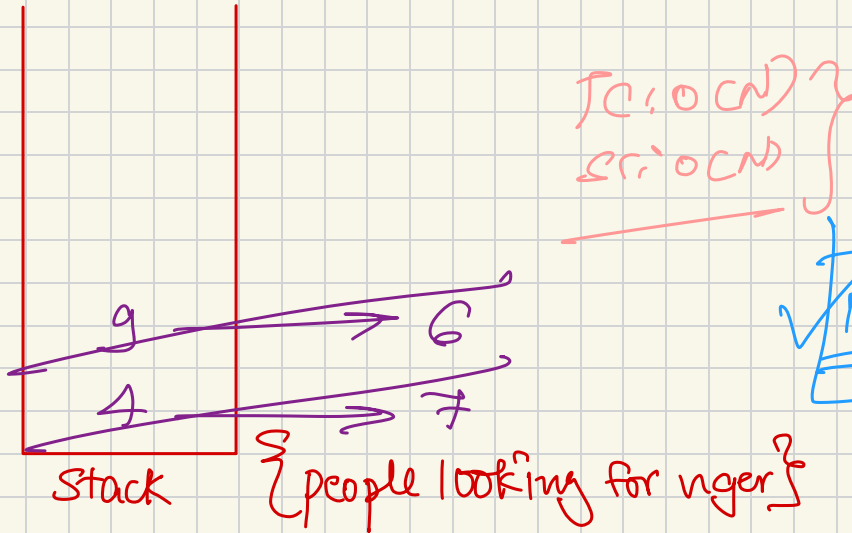
stack {potential nger}

nger

~~TC: O(N)  
SC: O(N)~~

## Approach 2

arr = { 3, 6, 1, 2, 7, 3, 4, 1, 2, 6 }  
↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑  
{ 6, 7, 2, 7, -1, 4, 6, 2, 6, -1 } → nger



✓ Monotonic Stack!



Approach 2 is better because

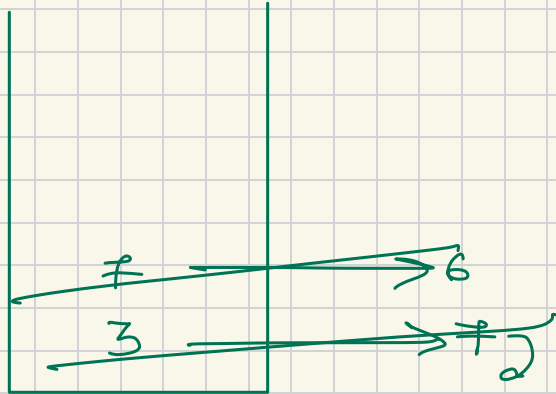
- ① it uses indexes
- ② Monotonic Stack

# Monotonic Stack

int[] arr = { 3, 4, 2, 7, 5, 4, 2, 6 }

0	1	2	3	4	5	6	7
3	4	2	7	5	4	2	6
↑	↑	↑	↑	↑	↑	↑	↑
4	7	7	-1	6	6	6	-1
-1	-1	4	-1	7	5	4	7

→ nges }  
→ ngel }



## H.W

- o Next greater Element of left
- o Next smaller Ele on left / right
- o Dry Run Monotonic Stack