PW SKILLS

Stacks -2

Ques:

Q1 : Balanced Brackets

Ques:

Q2 : Remove Consecutive Duplicates in a string

Ques:

Q3 : Next greater element

Ques:

Q4 : Previous greater element

eg1 { 10, 4, 2, 20, 40, 12, 30 }

o/p { -1 10 4 -1 -1 40 40 }

eg2 { 10, 20, 30, 40 }

{ -1, -1, -1, -1 }

Initialize = -1

ans[0] = -1

Naive :- 1) Take 2 loops, create a ans array

2) for (i = 1, ... n-1)

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

{ if arr[i] < arr[j]

} ans[i] = arr[j];

TC: O(N^2)

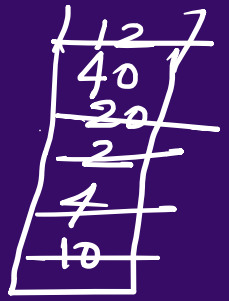
SC: O(1)

Better approach : stack.

1) Create a stack.
2) ans[0] = -1.
3) Push arr[0] in stack.
4) When seeing current element, we see top of stack & pop elements <= current element
5) If stack becomes empty, ans = -1
   otherwise ans = top of the stack.
6) Push current element in stack.

10, 4, 2, 20, 40, 12, 30

(-1 10 4 -1 -1 40 40)



TC: O(N)

SC: O(N) → stack space

Ques:

Q5 : Stock Span problem

Input - {100,80,60,70,60,75,85}

Output - {1 , 1 , 1 , 2 , 1 , 4 , 6}

10 4 5 90 120 80

1 1 2 4 5 1

PW SKILLS

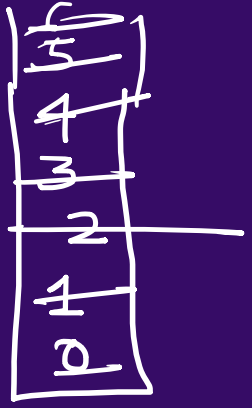
Naive approach :

1) For every element being visited, traverse elements on the left of it & increment the span while elements on left side are smaller.

TC: $O(N^2)$

SC: $O(1)$

	100	80	60	70	60	75	85	200
Day	0	1	2	3	4	5	6	7
Ans	1	1	1	2	1	4	6	8



1) Create a stack & push 0 in it.
2) set the answer of day 1 as 1 & run a loop
3) While stack is not empty & price of st.top <= price of current day, pop out the stack top
4) set ans of current day
   if stack is empty it1
   else i - st.top()
5) Push current day in the stack.
6) Print ans.

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

Ques:

Q6 : Largest Rectangle in Histogram

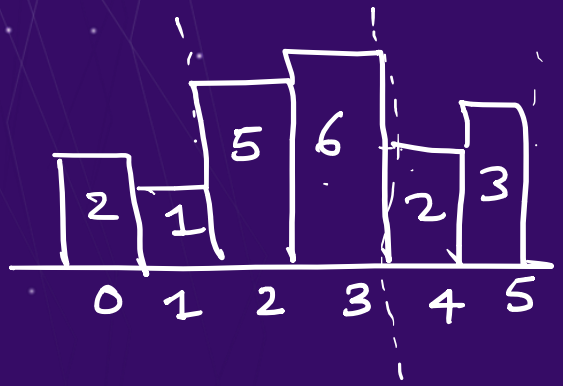
height 5 = 5 x 2 = 10 ✓

height 2 = 2 x 4 = 8

6 = 6 x 1 = 6

1 = 1 * 6 = 6

[Leetcode - 23]



3 ← next smaller Element

2 ← previous smaller Element

height * (NSE - PSE + 1)

5 * 2 = 10

6 * (3 - 3 + 1) = 6

2 * (5 - 2 + 1) = 8

[Leetcode - 23]

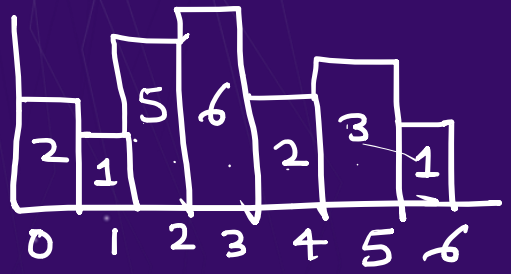
Naive approach:

1) Create 2 array PSE & NSE
2) For each element, find its PSE & NSE (N^2)
3) Max area = Max(height * (NSE - PSE + 1))

return area.

[Leetcode - 23]

Left smaller using stack



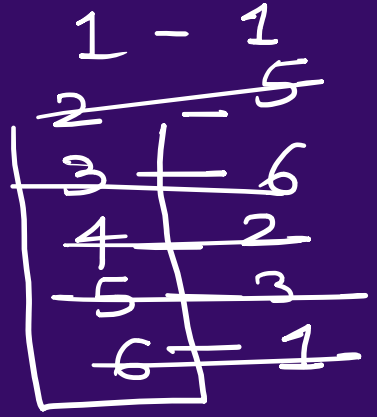
LSE

2	1	5	6	2	3	1
0	0	2	3	2	5	0
0	1	2	3	4	5	6

RSE

0	6	3	3	5	5	6
0	1	2	3	4	5	6
2	7	0	6	8	3	7

RSE



(RSE - LSE + 1) * height

width.

[Leetcode - 23]

TC: $O(N)$

SC: $O(N) + O(N) + O(N)$

↑ stack ↑ LSE ↑ RSE

$= O(N)$.

[Leetcode - 23]

THANK YOU