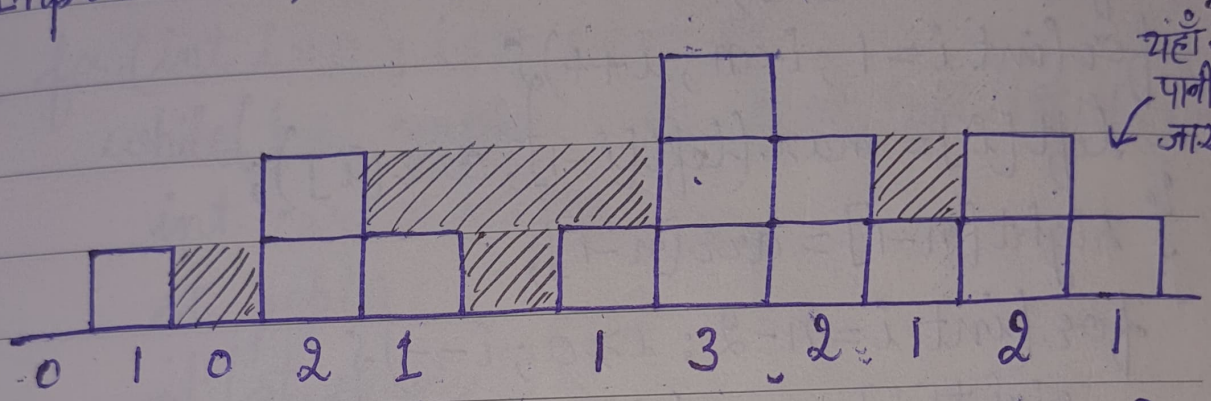
Trapping Rain Water

Que Given non-negative integers representing an elevation map where the width of each bar is 1, compute how much water it can trap after raining.

Input = 0, 1, 0, 2, 1, 0, 1, 3, 2, 1, 2, 1



Left[0] = arr[0] and Right[n-1] = arr[n-1]

हमें निकालना है कि water कहाँ store हो रहा है तो हम एक index पर खड़े रहकर उसके left right check करेंगे कि उससे बड़े height के elements है या नहीं। अगर होगे बड़े तो water store होगा वरना flow down कर जाएगा।

Array :	0	1	0	2	1	0	1	3	2	1	2	1
Left :	0	1	1	2	2	2	2	3	3	3	3	1
Right :	3	3	3	3	3	3	3	3	2	2	2	1
Minimum of LR :	0	1	1	2	2	2	2	3	2	2	2	1
ans :	0	0	1	0	1	2	1	0	0	1	0	0

Total ans = add all
=> 1+1+2+1+1
= 6

where ans = minimum of LR - arr[i]

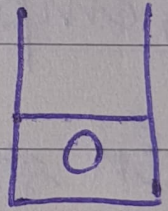
Dry run this code

inputs =	0	1	0	2	1	0	1	3	2	1	2	1	
index =	0	1	2	3	4	5	6	7	8	9	10	11	

Step-1 $i = 0$, empty stack ✓

push $i = 0$ in stack.

$i++;$

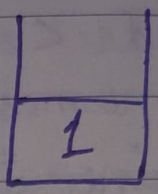


step-2 $i=1$, empty() X and ~~a[st.push~~ $a[st.top()] \leq a[i]$

curr = 0 $a[0] \leq a[1]$

Pop करदो 0 को $0 < 1$ ✓

stack is empty now ✓



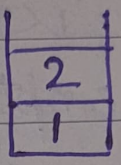
so Push $i=1$, i++;

Step-3 $i=2$, empty() X and $a[1] \leq a[2]$

$1 < 0$ X

so Push $i=2$ in stack

i++;



Step-4 $i=3$; empty() X and $a[2] < a[3]$

$0 < 2$ ✓

curr = 2, 2 pop होगया

empty X so

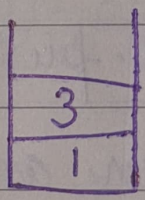
diff = 3 - 1 - 1 = 1

ans += $(\min(a[top()], a[i]) - a[curr]) * diff$.

$= (\min(a[1], a[3]) - a[2]) * 1$

$= (\min(1, 2) - 0)$

$= 1$

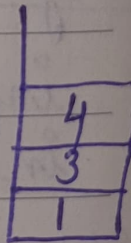


Push $i=3$ in stack, i++;

Step-5 $i=4$, empty() X and $a[3] < a[4]$

$2 < 1$ X

push $i=4$ in stack; i++;

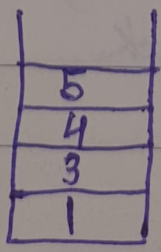


Step-6 $i=5$, empty() X and $a[4] < a[5]$

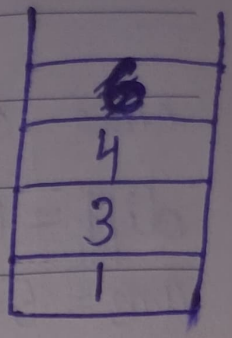
$1 < 0$ X

Push $i=5$ in stack;

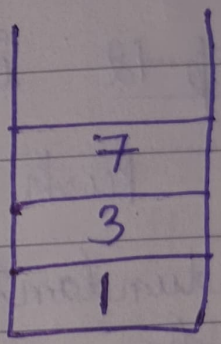
i++



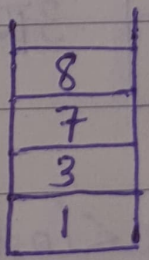
Step-7 i=6, empty()X and a[5]<a[6]
0<1 ✓
curr=5, 5 pop होगया
diff=6-4-1=1
ans=1+(min(a[4],a[6])-a[5])*diff
=1+(1-0)*1=2
Push i=6; i++;



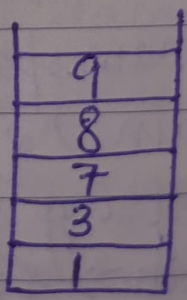
Step-8 i=7, empty()X and a[6]<a[7]
1<3 ✓
curr=6; 6 pop होगया
diff=7-4-1=2
ans=2+(min(arr[4],a[7])-a[6]);
=2+(1-1)*2=2+0*2=2
empty X, a[4]<a[7]=1<3 ✓
Now, curr=4 and 4 pop होगया
diff=7-3-1=3
ans=2+(min(arr[3],arr[7])-arr[6])*diff
=2+(2-1)*3=2+1*3=2+3=5
Push i=7 and i++;



Step-9 i=8
empty X and a[7]<a[8]
3<2 X
Push i=8; i++;



Step-10 i=9 empty X and a[8]<a[9]
2<1 X
Push i=9; i++;



Step-11 i=10 ; empty() ✗

a[9] < a[10]

1 < 2 ✓

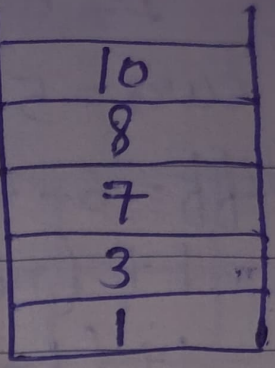
curr = 9, 9 popped

diff = 10 - 8 - 1 = 1 ;

ans = 5 + (min(a[8], a[10]) - arr[9]) * 1

= 5 + (2 - 1) * 1 = 5 + (1) = 6

Push i=10 ; i++



Step-12 i=11 ; empty() ✗ and a[10] < a[11]

Push i=11 ; ✗ 2 < 1