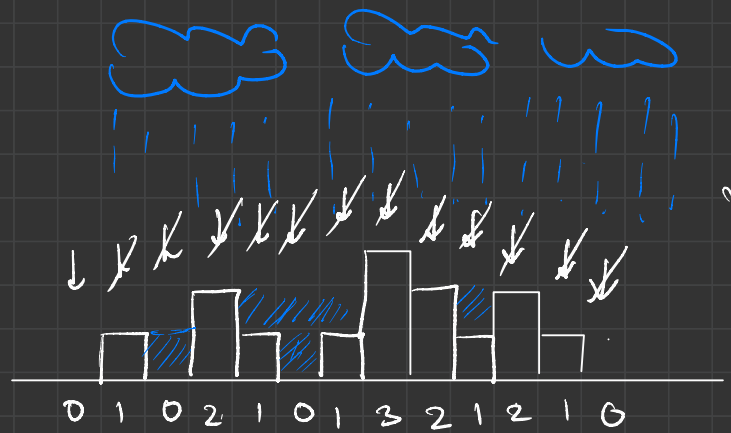
Trapping Rain Water



max H = ~~0~~ ~~1~~ ~~2~~ 3

0	1	1	2	2	2	2	3	3	3	3	3	3
3	3	3	3	3	3	3	3	2	2	2	1	0

Lmax

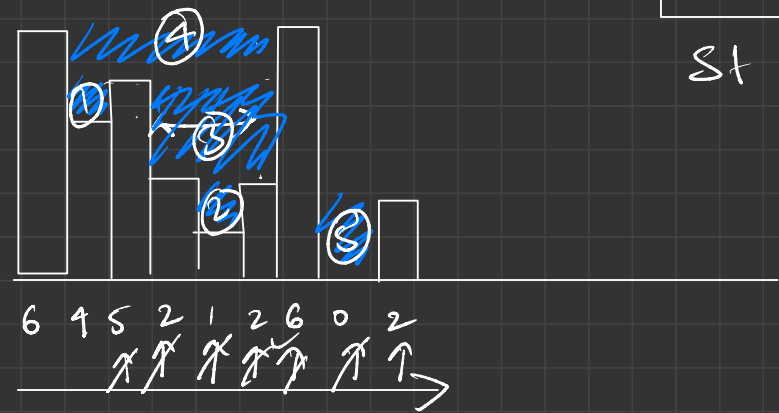
O(N)

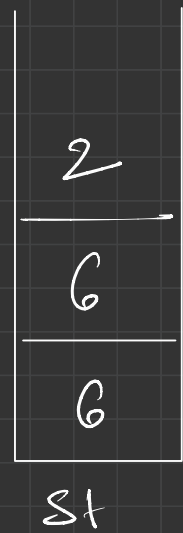
Rmax

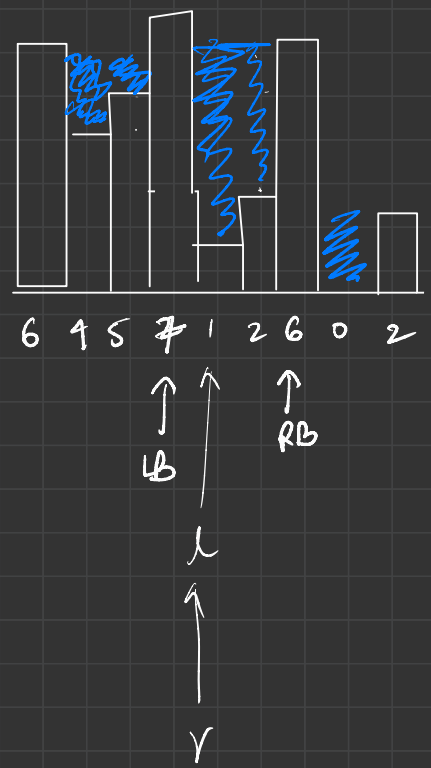
O(N)

nger >=

ngel >








```
if (LB <= RB)
{ if (arr[l] < LB)
  { water = [LB - arr[l]] x1;
    l++;
  } else
  { LB = arr[l]; l++; }
}
else
{ if (arr[r] < RB)
  { water = [RB - arr[r]] x1
    r--;
  } else
  { RB = arr[r]; r--; }
}
```

min Stack.

push(4)

push(1)

push(3)

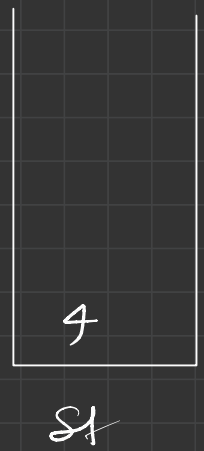
$\boxed{\text{getMin()}}$ → 1

pop()

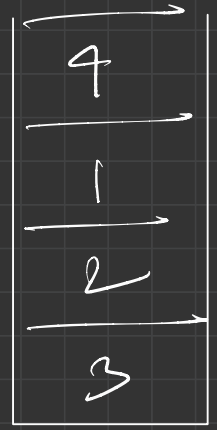
$\boxed{\text{getMin()}}$ → 1

pop()

getmin() → 4



getMin() → O(N)



st

temp

order preserved

min = ~~∞~~ ~~5~~ ~~4~~ 1

push(4)
push(3)
push(8)
push(6)
getMin()
pop()
pop()
pop()
getMin()

pair
val, minvalue

push
TC: O(1)

pop
TC: O(1)

getMin
TC: O(1)

1 -2

-2
6
8
3
4

minele = ~~4~~ ~~3~~ 1

newmin

$\boxed{\text{val} = x - \text{minvalue};}$

stack

prevmin

prev M
= 1 − (−2)
= 3

Sum of Subarray Minimums

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

3 + 2 + 2 + 1 + 1 + 1

(3), (3,2), (3,2,4), (3,2,4,1), (3,2,4,1,5), (3,2,4,1,5,2)

2 + 2

(2) (2,4) — — —

$O(N^2) \Longrightarrow$ Brute

	0	1	2	3	4	5
arr[] =	3	2	4	1	5	2
nseli	-1	-1	1	-1	3	3
nseri	1	3	3	6	5	6

(1)
(4,1)
(2,4,1)
(3,2,4,1)

7×3

nL

idx − (nseli)

no. of Ele left

add(nL × nR × value)

nR

nseri − idx

6 − 3 = 3

12×1

add total

arr[] = {4, 3, 4, 2, 5}

(4) → 1×4 = 4

(3), (3,4), (4,3), (4,3,4) → 4×3 = 12

(4) → 1×4 = 4

2×2 = 16

1×5 = 5

4 + 12 + 4 + 16 + 5

=

(2), (2,5), (4,2), (4,2,5),

(3,4,2), (3,4,2,5)

(4,3,4,2), (4,3,4,2,5)