Reverse Integer.

$x = 1200$
$rev = 21$

int number 32 bit

↓

reverse this number → $[-2^{31}, 2^{31}-1]$

↓

0

$x = \overset{\downarrow}{4}\overset{\downarrow}{5}\overset{\downarrow}{3}\overset{\downarrow}{2}$

↳ reverse : 2354

$x \% 10 = 2$
$x / 10 = 453$

✓ Agenda

- Subarray sum minimums
- trapping rain water
- Min Stack

Subarray Sum Minimums

$$\text{int[] arr} = \left\{\begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \\ 3, & 2, & 4, & 5, & 1, & 8 \end{array}\right\}$$

(3) → 3

(3, 2) → 2

(3, 2, 4) → 2

(3, 2, 4, 5) → 2

(3, 2, 4, 5, 1) → 1

(3, 2, 4, 5, 1, 8) → 1

(2) → 2

(2, 4) → 2

(2, 4, 5) → 2

(2, 4, 5, 1) → 1

(2, 4, 5, 1, 8) → 1

(4) → 4

(4, 5) → 4

(4, 5, 1) → 1

(4, 5, 1, 8) → 1

(5) → 5

(5, 1) → 1

(5, 1, 8) → 1

(1) → 1

(1, 8) → 1

(8) → 8

✓ $\boxed{\Sigma_i = 46}$

Brute force.

- get all the Subarrays, identify min in them
- add all min.

```
int sum = 0;
for (int i = 0 ⟶ n)
{
    int min = +∞;
    for (int j = i ⟶ n)
    {   min = Math.min(min, arr[j]);
        sum += min;
    }
}
```

TC : $O(N^2)$
SC : $O(1)$

$$\text{int[] arr} = \left\{\begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \\ 3, & 2, & 4, & 5, & 1, & 8 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ a & b & c & d & e & f \end{array}\right\}$$

$$\sum \text{subarray min} = a \times 3 + b \times 2 + c \times 4 + d \times 5 + e \times 1 + f \times 8$$

0 1 2 3 4 5

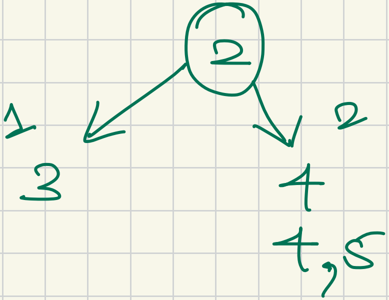
int[] arr = {3, 2, 4, 5, 1, 8}

people you can include from right

total Subarrays = 1 + nsr + nsl + nsr x nsl

coming alone

people you can include from left.



$1 \times 2 = \boxed{2}$

3 2 4

3 2 4 5

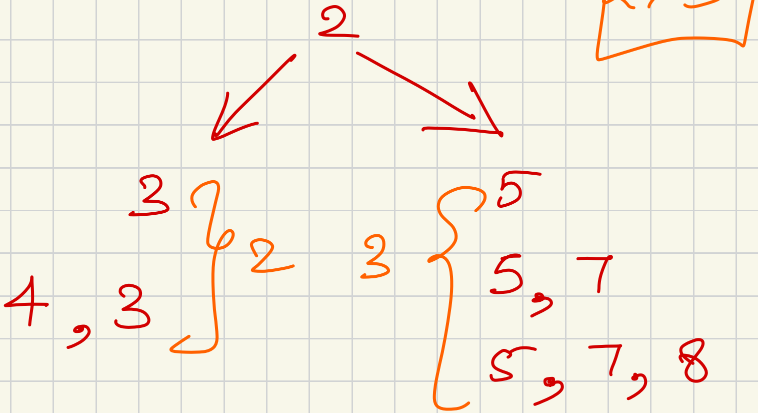
4, 3, 2, 5, 7, 8

↑

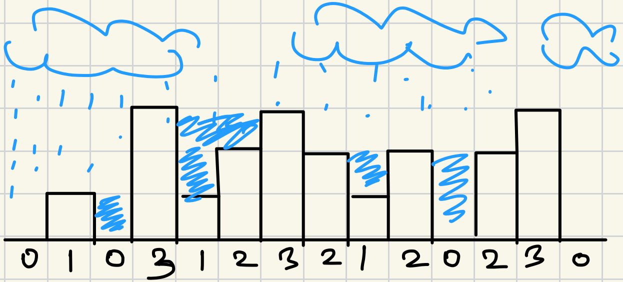
total Subarray = 1 + 3 + 2 + 2x3 ✓

$\boxed{2\times 3}$

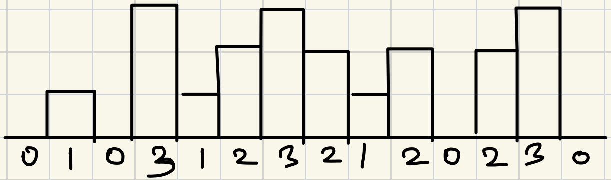
2

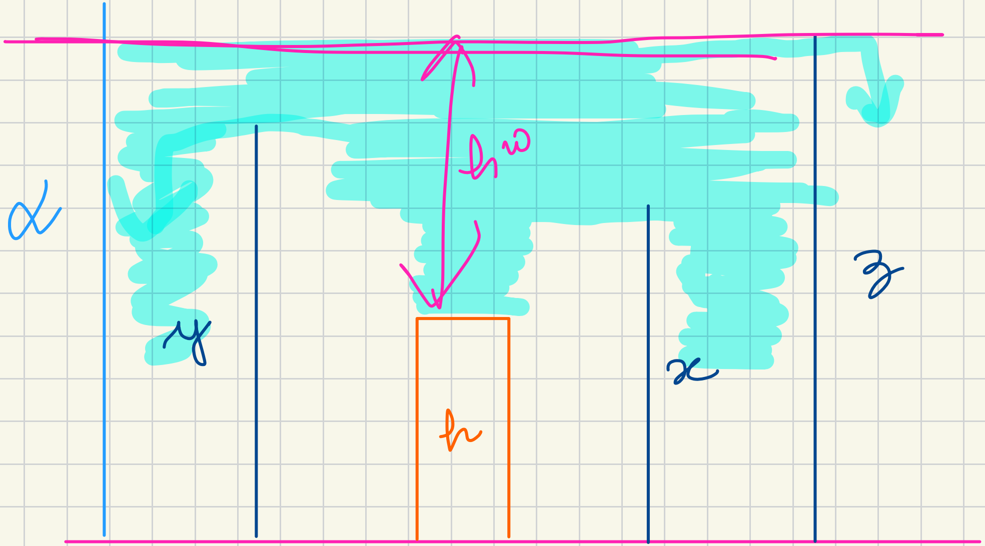


Trapping Rain Water

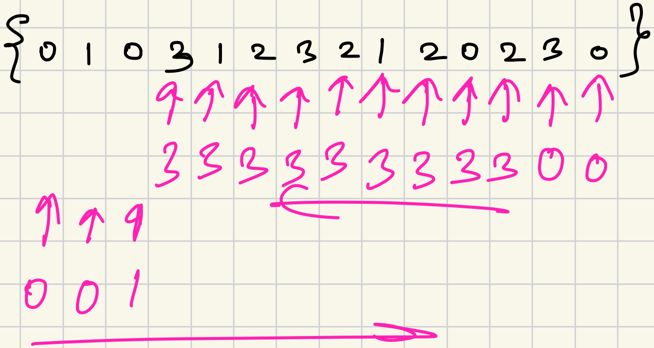


total Water = 7

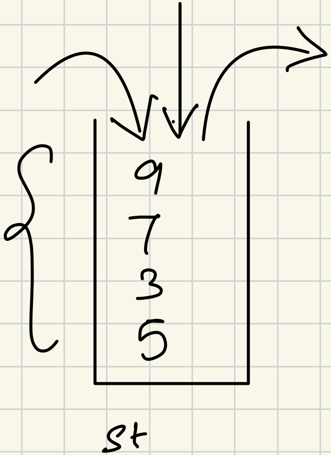




h of water = min{tallest r, tallest l} − h of Building


{0 1 0 3 1 2 3 2 1 2 0 2 3 0}
3 3 3 3 3 3 3 3 3 0 0
0 0 1

Min Stack



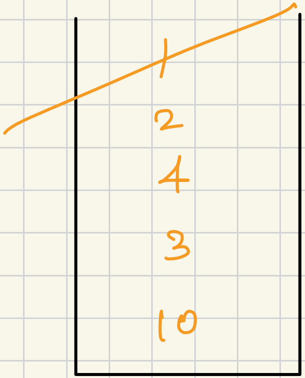
st.getMin() → returns min value in stack

→ min

→ TC: O(1)

pre = ~~3~~ 2

minVal = ~~+∞~~ ~~10~~ ~~3~~ ~~2~~ ~~1~~ 2



Stack

push(10)

push(3)

push(4)

push(2)

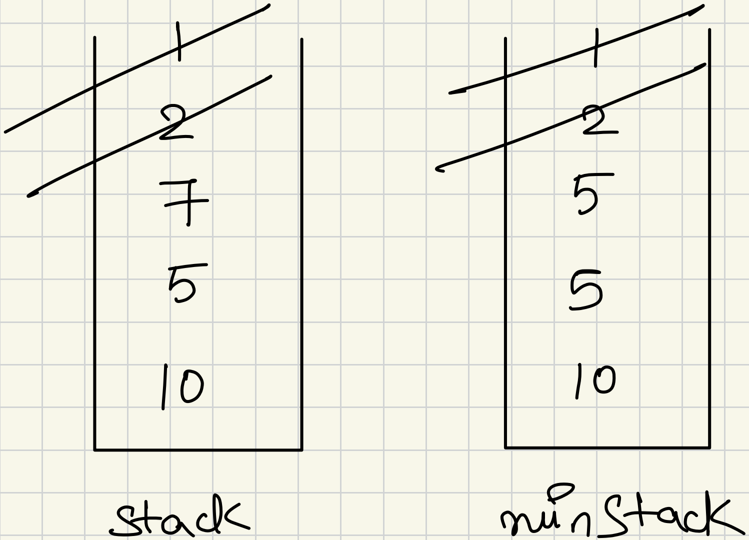
push(1)

getMin() → 1

pop()

not work!

Approach 1

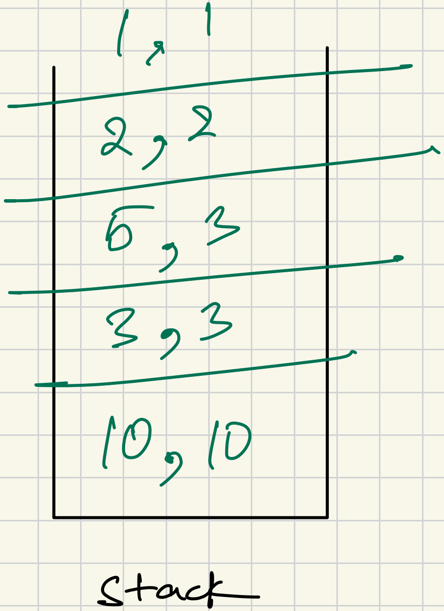


minVal = ~~+∞~~ ~~10~~ ~~5~~ ~~2~~ 1

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

Approach 2

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



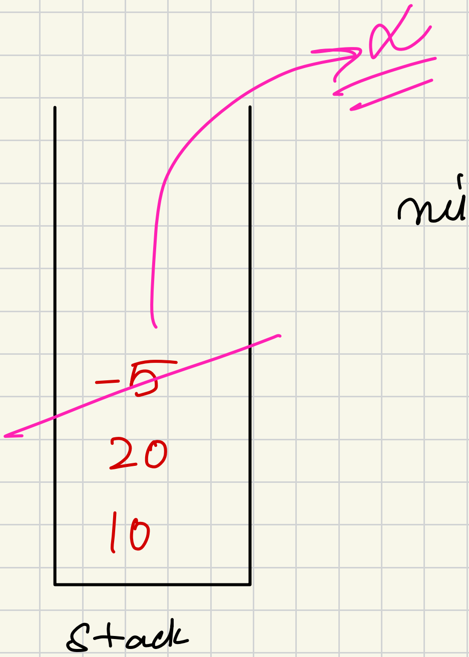
minVal = ~~+∞~~ ~~10~~ ~~3~~ ~~2~~ 1

```
class Pair
{
    int value;
    int min;
}
```

Approach 3

TC : O(N) SC : O(1)

prevMin > currMin
x



α

minVal = ~~10~~ 5 ~~3~~

$\alpha = x - \text{prevMin}$

$1 - 3$

$\text{prevMin} = x - \alpha$

$= 1 - (-2)$

$= 3$

push(10)
push(20)
push(5)
push(7)
push(3)
push(1)

getMin() → 1
pop() ⇝ 1
getMin() ⇝ 3
pop() ⇝ 3
pop()

$= 3 - (-2)$

$= 5$

$= 5 - (-5)$

$= \boxed{10}$

H·W

① reverse substring b/w each pair of paranthesis