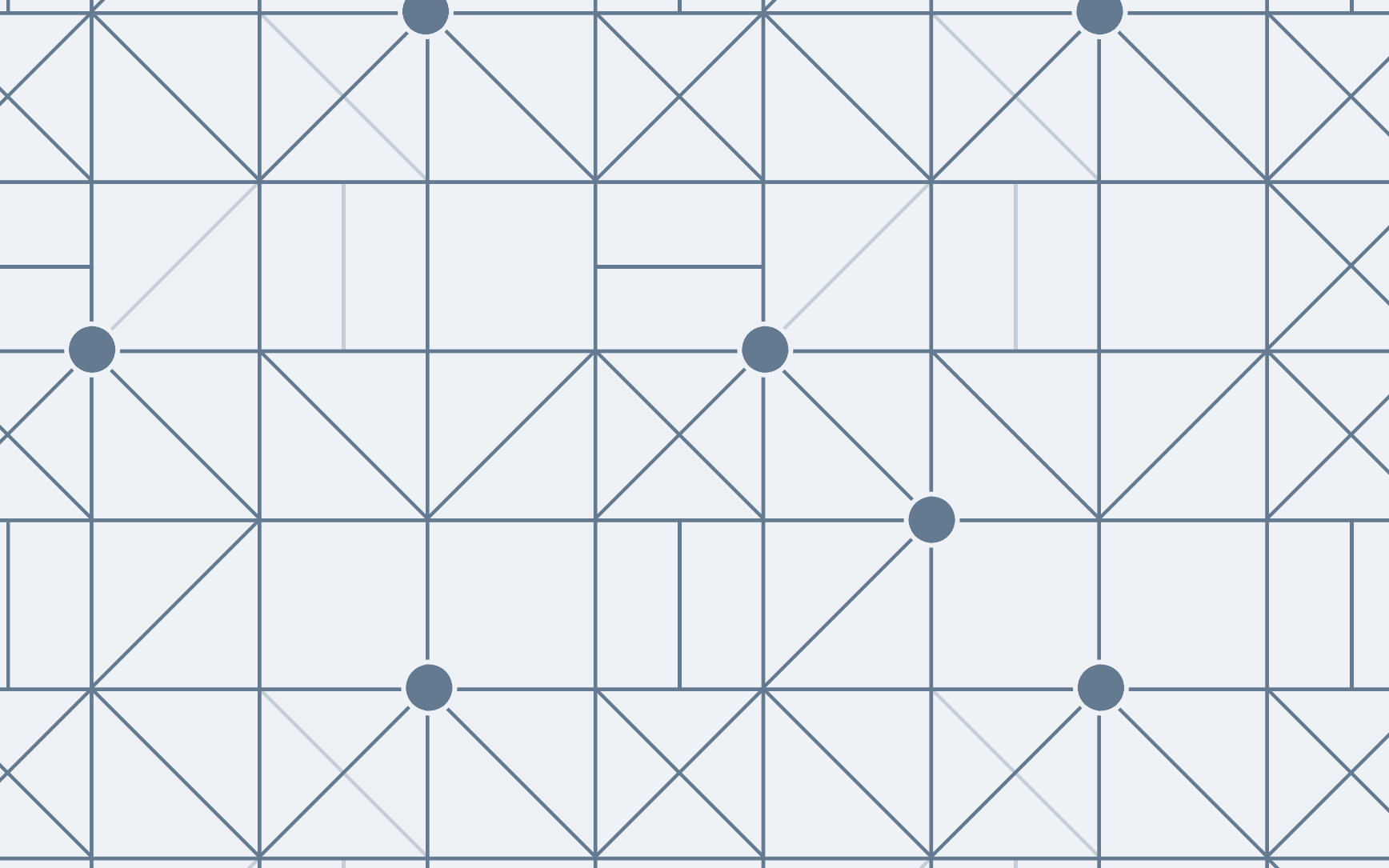
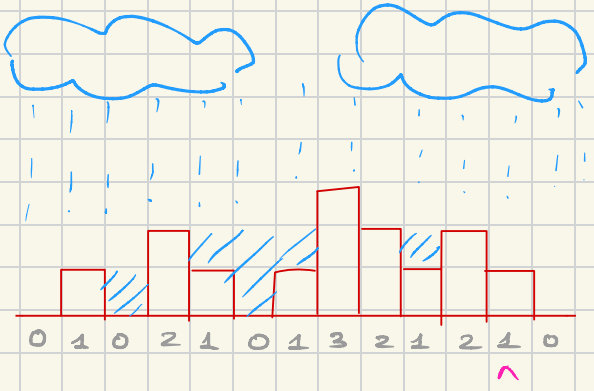


Agenda

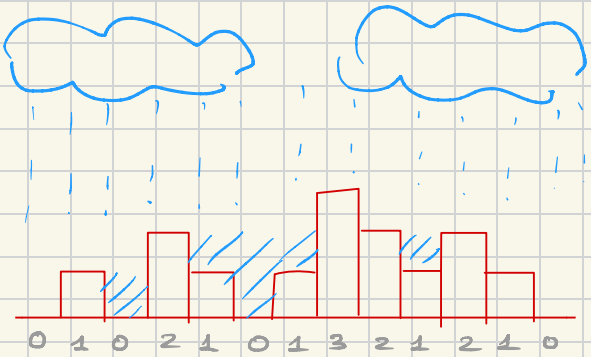
1. Trapping Rain water
2. Sum of Subarray minimums
3. Minimum Stack

Trapping Rain Water.

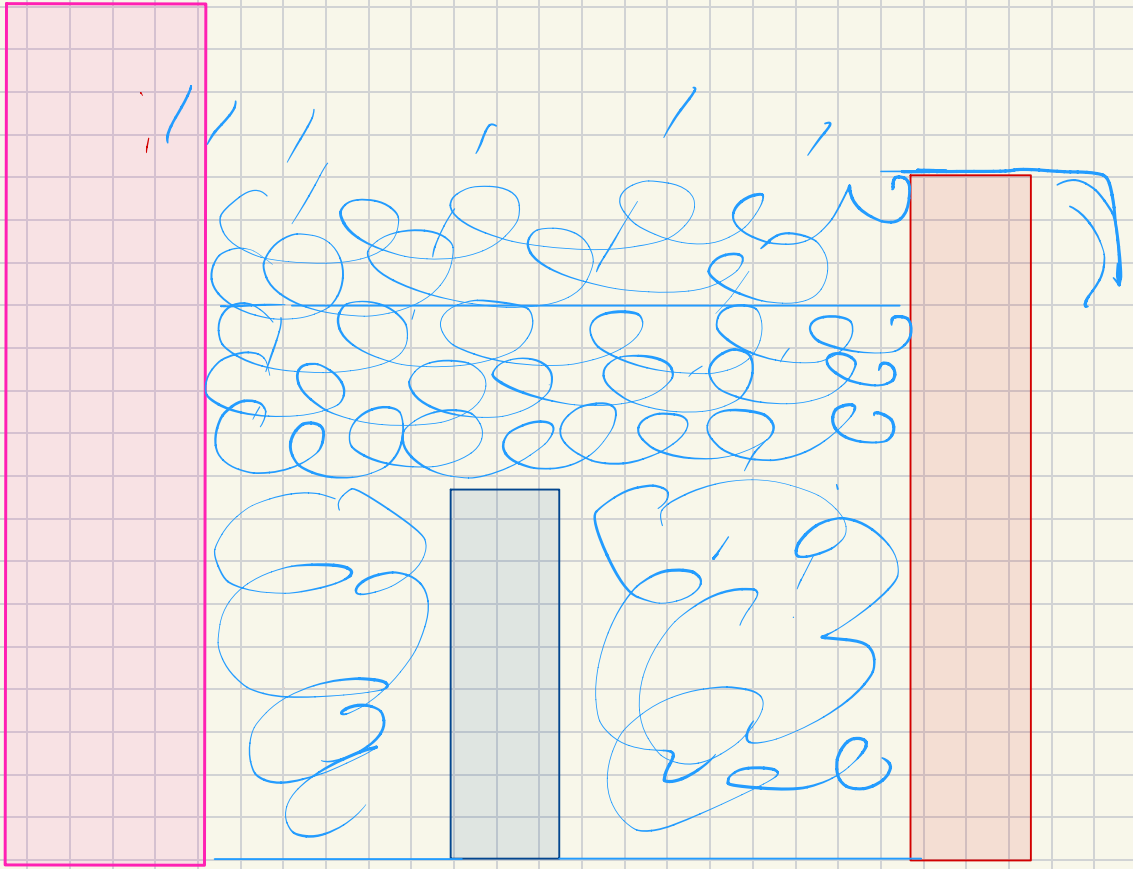


Calculate the trapped rain water.

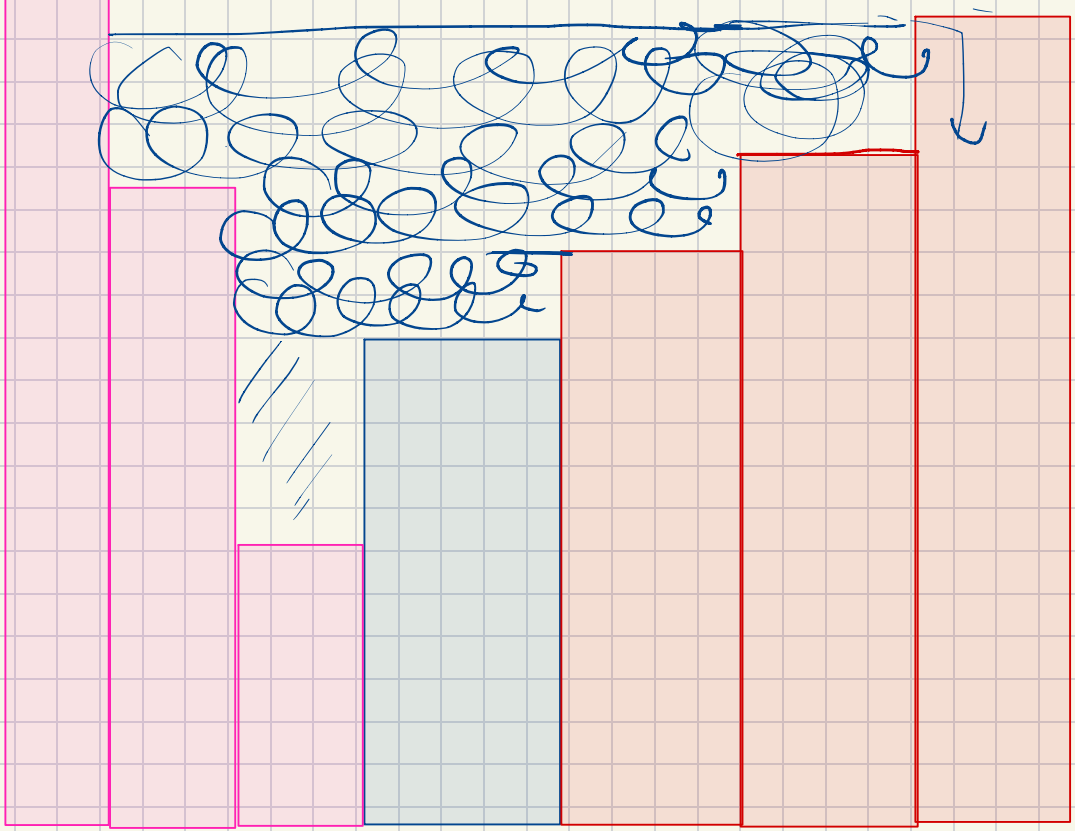
ans = 6 units of water

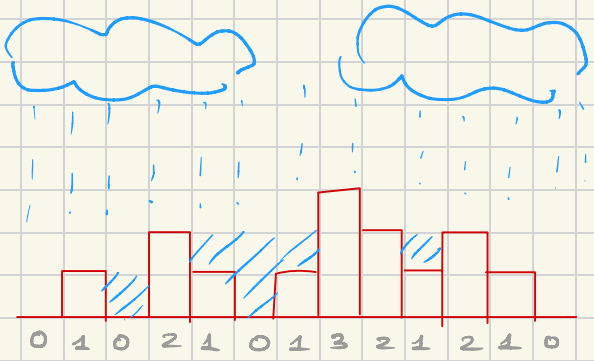


total water = $\sum$ Water above each building



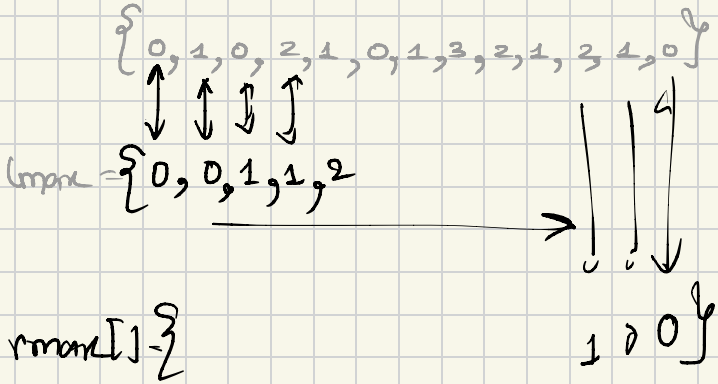
min(max on right, max on left).





lmax[]	0	0	1	1	2	2	2	2	3	3	3	3	3
rmax[]	3	3	3	3	3	3	3	2	2	2	1	0	0
height	0	0	1	1	2	2	2	2	2	2	1	0	0
till water can go	0	0	1	0	1	2	1	0	0	1	0	0	

$\sum$ = 6 unit ✓


{0, 1, 0, 2, 1, 0, 1, 3, 2, 1, 3, 1, 0}
tmpArr = {0, 0, 1, 1, 2
rmArr[] = {
1 > 0}

Sum of subarray minimums

Given arr = {3, 2, 4, 1, 5, 2, 1}

{3}(3) {2}(2) {4}(4) {1}(1) {5}(5) {2}(2) {1}(1)

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

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

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

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

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

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

$\sum$ = 43

↳ ans!

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

Brute force

1. Calc. all subarray min
2. Sum it up

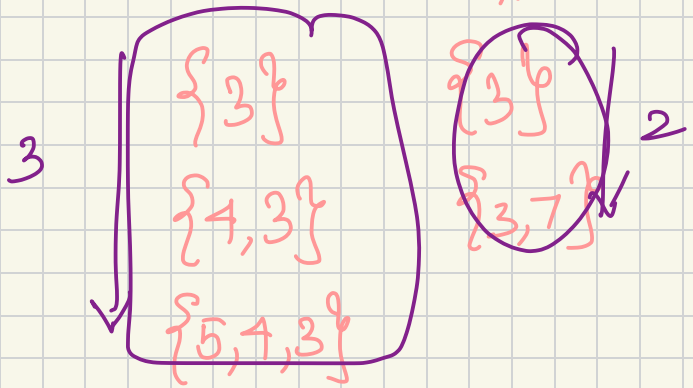
} TC: O(N²), SC: O(1)

```
int Sum=0;
for (i = 0 → n)
    int minValue = ∞;
    for (j = i → n)
        min(minValue, arr[j]);

    Sum += minValue
```

int[] arr = {2, 5, 4, 3, 7, 1, 8}

0 1 2 3 4 5 6



3 x 2 = 6 ✓ Subarrays where 3 is min^m

Sum = 6 x 3 ✓

int[] arr = { 3, 2, 4, 1, 5, 2, 1 }
(0 1 2 3 4 5 6)

nseli = { -1, -, 1, -1, 3, 3, 3 }

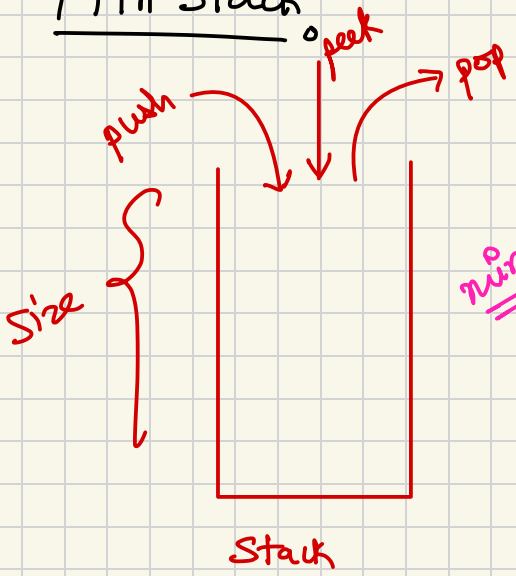
nseri = { 1, 3, 3, 7, 5, 6, 7 }

x number of ele on left = (index - nseli)

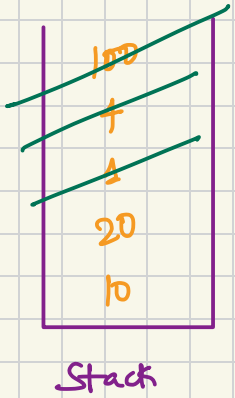
y number of ele on Right = (nseri - index)

$$ans = \sum_{i=0}^{n-1} x_i \times y_i \times arr[i]$$

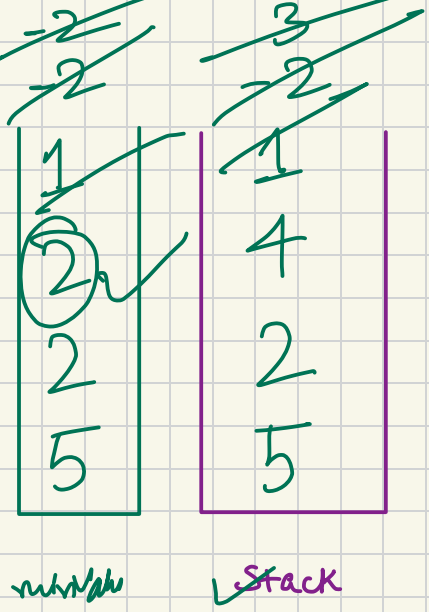
Min Stack



minStack()
↳ TC: O(1)



st.getMin() → 1
st.getMin() → 10

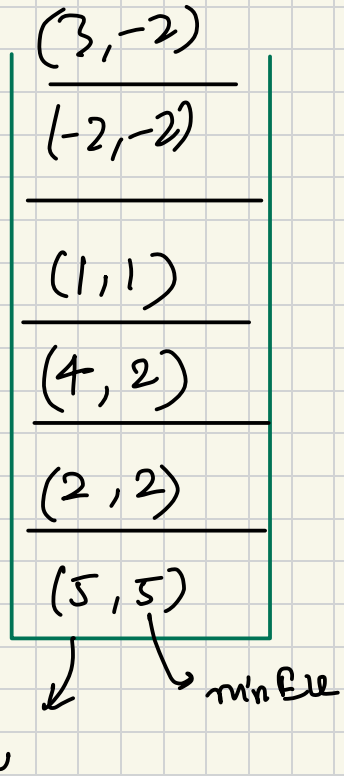


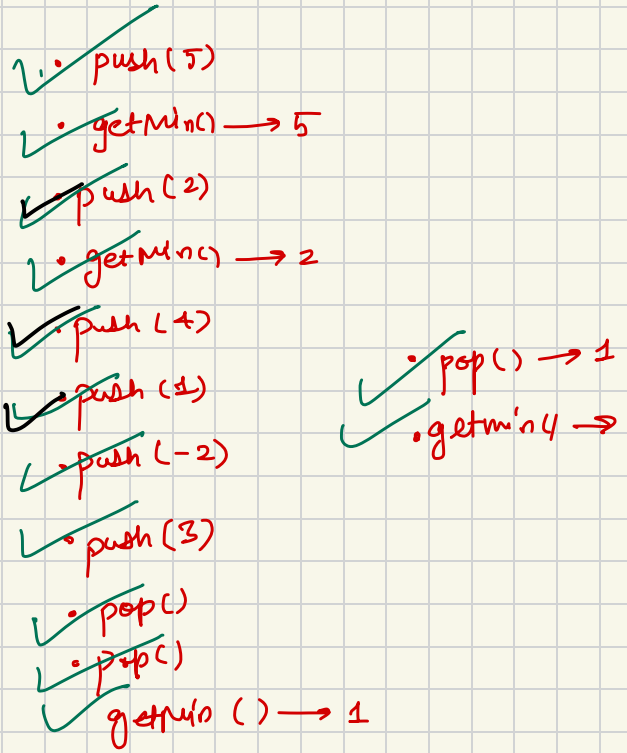
min = ~~2~~ 1

- push(5)
- getMin() → 5
- push(2)
- getMin() → 2
- push(4)
- push(1)
- push(-2)
- push(3)
- pop()
- pop()
- getMin() → 1

- pop() → 1
- getmin() → 2

4
2
5


(3,-2)
(-2,-2)
(1,1)
(4,2)
(2,2)
(5,5)
minEle
val


push(5)
getMin() → 5
push(2)
getMin() → 2
push(4)
push(1)
push(-2)
push(3)
pop()
pop()
getMin() → 1
pop() → 1
getMin() →