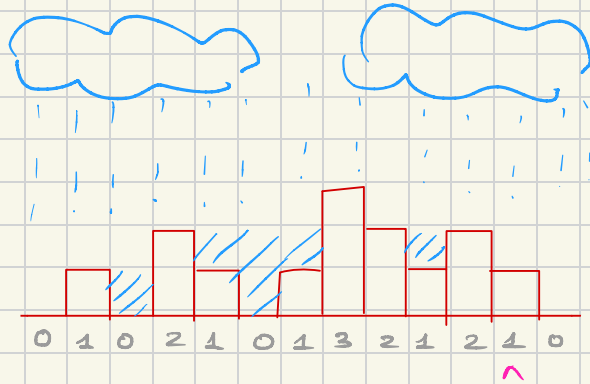




## Agenda

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

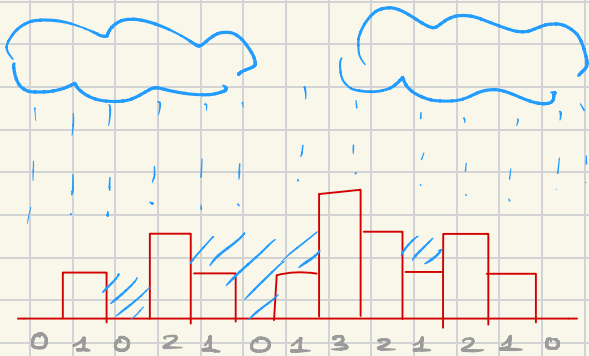
## Trapping Rain Water



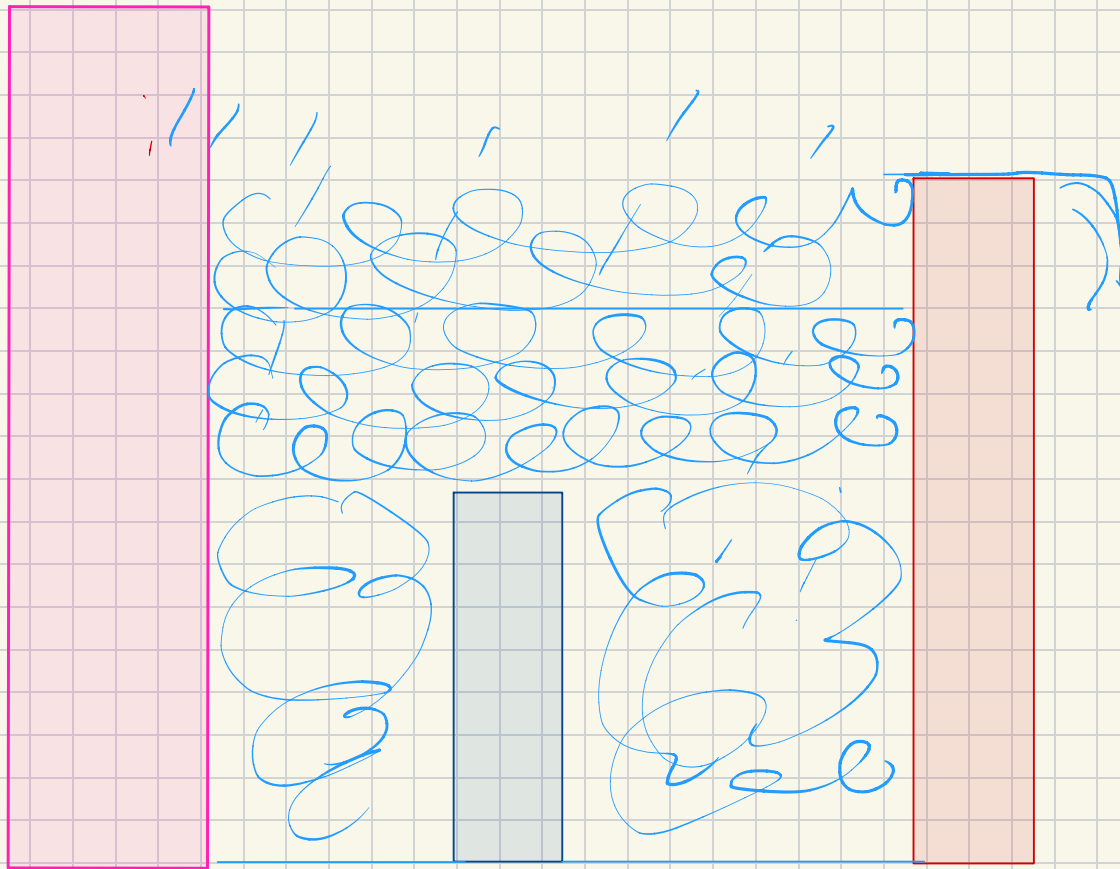
heights[] =

Calculate the trapped rain water!

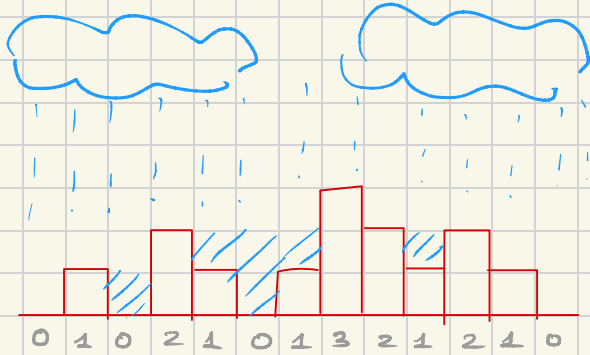
ans = 6 units of water



$$\text{total water} = \sum \text{Water above each building}$$



Hand-drawn diagram of a neural network architecture on grid paper. The diagram shows a sequence of layers: an input layer (pink), a hidden layer (blue), and an output layer (red). Above the hidden layer is a layer of blue circles representing nodes. A yellow box with diagonal lines is labeled "Bias".



lmax[] 0 0 1 1 2 2 2 2 3 3 3 3 3

rmax[] 3 3 3 3 3 3 2 2 2 2 1 0 0

height 0 0 1 1 2 2 2 2 2 2 1 0 0

till  
 water  
 can go

$$\sum 0 0 1 0 1 2 1 0 0 1 0 0 0 = 6 \text{ unit} \checkmark$$

$\{0, 1, 0, 2, 1, 0, 1, 3, 2, 1, 2, 1, 0\}$

$\updownarrow \updownarrow \updownarrow \updownarrow$

$\text{map} = \{0, 0, 1, 1, 2\}$

$\downarrow \downarrow \downarrow$   
 $1 \ 0 \ 0$

$\text{rmax}[1] = \{$



# Sum of subarray minimums

ex: arr = { 3, 2, 4, 1, 5, 2, 1 }

$\{3\}$   $\{2\}$   $\{4\}$   $\{1\}$   $\{5\}$   $\{2\}$   $\{1\}$   
 $\{3, 2\}$   $\{2, 4\}$   $\{4, 1\}$   $\{1, 5\}$   $\{5, 2\}$   $\{2, 1\}$   
 $\{3, 2, 4\}$   $\{2, 4, 1\}$   $\{4, 1, 5\}$   $\{1, 5, 2\}$   $\{5, 2, 1\}$   
 $\{3, 2, 4, 1\}$   $\{2, 4, 1, 5\}$   $\{4, 1, 5, 2\}$   $\{1, 5, 2, 1\}$   
 $\{3, 2, 4, 1, 5\}$   $\{2, 4, 1, 5, 2\}$   $\{4, 1, 5, 2, 1\}$   
 $\{3, 2, 4, 1, 5, 2\}$   $\{2, 4, 1, 5, 2, 1\}$   
 $\{3, 2, 4, 1, 5, 2, 1\}$

$$\sum = \underline{\underline{43}}$$

↳ ans!

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

## Brute-force

- ① Calc. all subarray min
  - ② Sum it up
- TC:  $O(N^2)$   
SC:  $O(1)$

```
int sum = 0;
```

```
for (i = 0 → n)
```

```
    int min value = ∞;
```

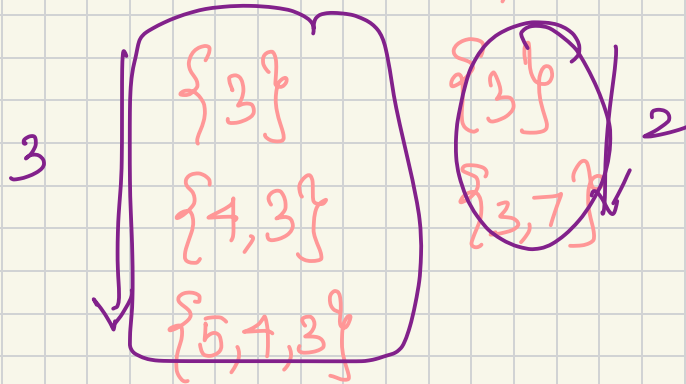
```
    for (j = i → n)
```

```
        min(min value, arr[j]);
```

```
    sum += min value
```

int[] arr = {<sup>0 1 2 3 4 5 6</sup>2, 5, 4, 3, 7, 1, 8}

*(A red double-headed arrow is drawn under the first three elements: 2, 5, 4)*



3 x 2 = 6 Subarrays where 3 is min

Sum = 6 x 3 ✓

$$\text{int[] arr} = \{ \overset{0}{3}, \overset{1}{2}, \overset{2}{4}, \overset{3}{1}, \overset{4}{5}, \overset{5}{2}, \overset{6}{1} \}$$

$$\text{nsel} = \{-1, -, 1, -1, 3, 3, 3\}$$

$$\text{nsel} = \{1, 3, 3, 7, 5, 6, 7\}$$

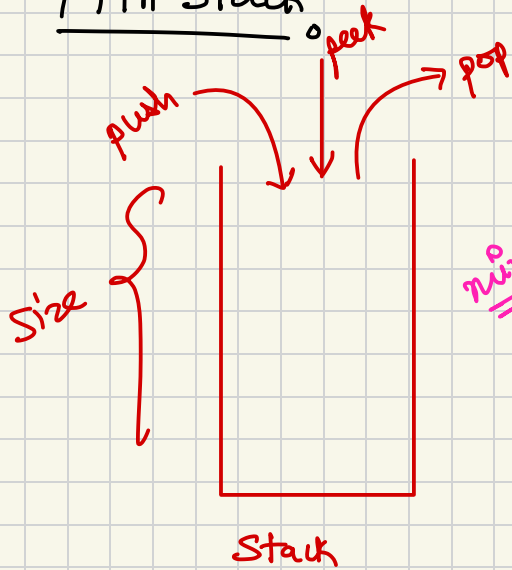
$$x \text{ number of ele on Left} = (\text{index} - \text{nsel})$$

$$y \text{ number of ele on Right} = (\text{nsel} - \text{index})$$

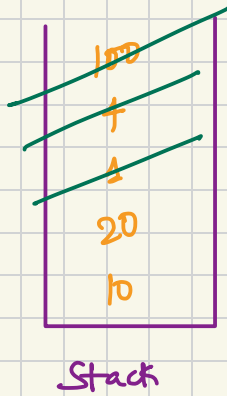
$$\text{ans} = \sum_{i=0}^{n-1} x_i \times y_i \times \text{arr}[i]$$

=====

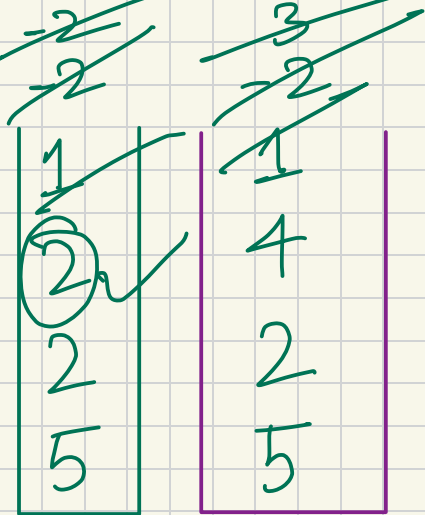
# Min Stack



minStack()  
↳ TC:  $O(1)$



$\left\{ \begin{array}{l} \text{st.getMin() } \rightarrow 1 \\ \text{st.getMin() } \rightarrow 10 \end{array} \right\}$



minStack

Stack

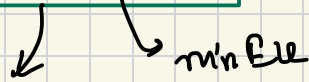
min = ~~0~~  
~~1~~

- ✓ • 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

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


  
 val                      minEle

✓ • push(5)

✓ • getMin() → 5

✓ • push(2)

✓ • getMin() → 2

✓ • push(4)

✓ • push(1)

✓ • push(-2)

✓ • push(3)

✓ • pop()

✓ • pop()

✓ • getMin() → 1

✓ • pop() → 1  
 ✓ • getMin() → 1