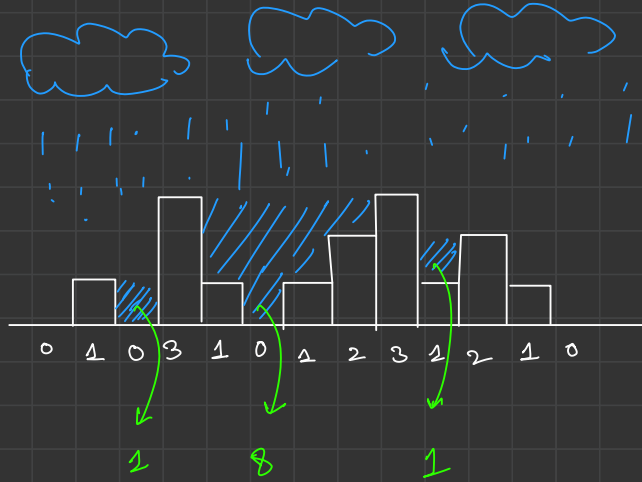




## Trapping Rain Water

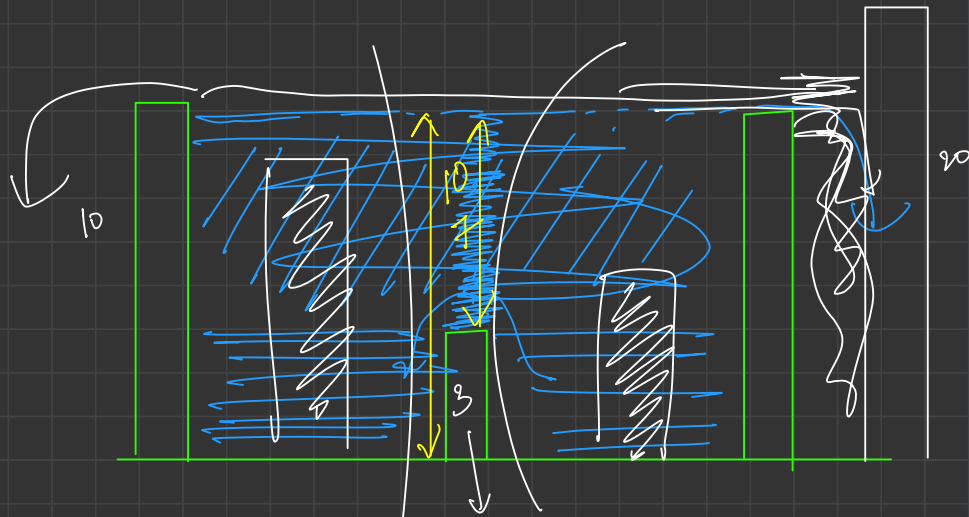


### Brute force

= 10 sq units of water

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

- ① go to each building
- ② get  $\text{max}^m$  on left and  $\text{max}^m$  on right
- ③ get limiting building =  $\min$  of curr Building +  $\min$  of  $w$
- ④  $h$  of  $w$  = limiting -  $h$  of building



largest  
on left

largest  
on right

among them the shorter building will become  
limiting building.

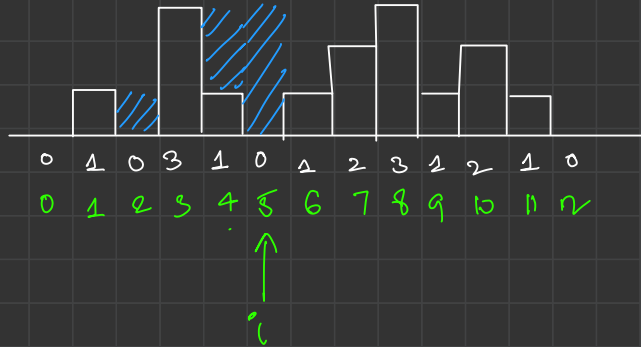
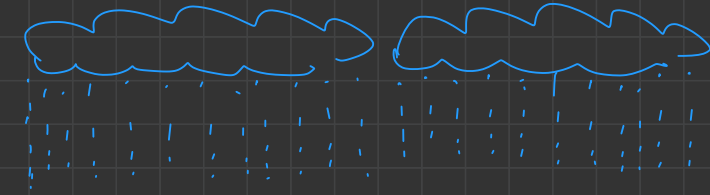
```
// TC: O(N^2), SC: O(1)
public void TappingWater(int[] arr, int n) {
    //Write code here and print output

    int amtOfWater = 0;
    for (int i = 0; i < n; i++) {
        int lmax = 0;
        // left side 0 -> i - 1
        for (int j = 0; j <= i - 1; j++) {
            lmax = Math.max(lmax, arr[j]);
        }

        int rmax = 0;
        // right side i + 1 -> n - 1
        for (int j = i + 1; j <= n - 1; j++) {
            rmax = Math.max(rmax, arr[j]);
        }

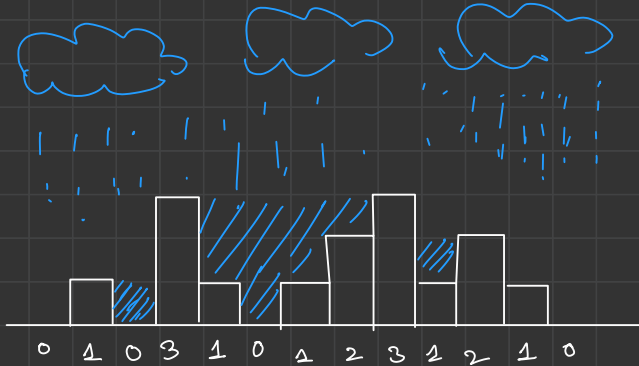
        // height of water plus building
        int totalHeight = Math.min(lmax, rmax);
        if (totalHeight > arr[i]) {
            int waterHeight = totalHeight - arr[i];
            amtOfWater += (waterHeight * 1);
        }
    }

    System.out.println(amtOfWater);
}
```



$lmax = 3$   
 $rmax = 3$

$totalHeight = 3$



max

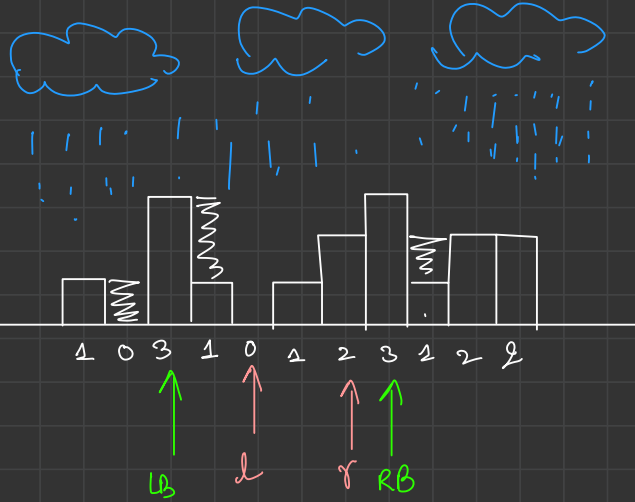
0 1 0 3 1 0 1 2 3 1 2 1 0

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

lmax

0 0 1 1 3 3 3 3 3 3 3 3 3

$$\left. \begin{aligned} & \text{rmax}[i] = \text{max}(\text{rmax}[i+1], \text{arr}[i+1]) \\ & \text{lmax}[i] = \text{max}(\text{lmax}[i-1], \text{arr}[i-1]) \end{aligned} \right\}$$



$$\begin{aligned} TC: O(N) \\ SC: O(1) \end{aligned}$$

$$\underline{\underline{while (l \leq r)}}$$

$$\underline{LB \leq RB}$$

$$\underline{\text{total height} = LB;}$$

}

update LB;

l++;

limiting

$$\begin{aligned} RB < LB \\ \underline{\text{total h} = RB;} \\ \left\{ \begin{aligned} &\text{update RB} \\ &r--; \end{aligned} \right. \end{aligned}$$

# Sum of Subarray minimums

arr[] : { 0 1 2 3 4 5 }  
          { 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           1           1           1  
      ↑        ↑        ↑        ↑        ↑  
{ 2 }   { 2, 4 }   { 2, 4, 1 }   { 2, 4, 1, 5 }   { 2, 4, 1, 5, 2 }

        1           1           1           1  
      ↑        ↑        ↑        ↑  
{ 4 }   { 4, 1 }   { 4, 1, 5 }   { 4, 1, 5, 2 }

        1           1           1  
      ↑        ↑        ↑  
{ 1 }   { 1, 5 }   { 1, 5, 2 }

        5           2  
      ↑        ↑  
{ 5 }   { 5, 2 }

        2  
      ↑  
{ 2 }

$$\sum_{\min} = \textcircled{36} \rightarrow \text{off}$$

## Bruteforce

get all subarray, generate min<sup>m</sup> and add.

TC:  $O(N^2)$

SC:  $O(1)$

arr[] : { 0 1 2 3 4 5 }  
          { 3, 2, 4, 1, 5, 2 }

ans = 36

TC:  $O(N)$ ?

3, 2, 4, 1

1, 5, 2



{1} {1, 5} {1, 5, 2}

{4, 1} {4, 1, 5} {4, 1, 5, 2}

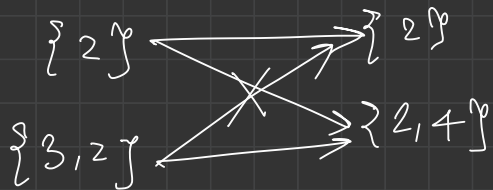
{2, 4, 1} {2, 4, 1, 5} {2, 4, 1, 5, 2}

{3, 2, 4, 1} ——— ———



arr[] : { <sup>0</sup>3, <sup>1</sup>2, <sup>2</sup>4, <sup>3</sup>1, <sup>4</sup>5, <sup>5</sup>2 }

2x2 = 4 Subarrays



{2} {2, 4} {3, 2}  
{3, 2, 4}

arr[] : { 0 1 2 3 4 5 }  
          { 3, 2, 4, 1, 5, 2 }

nsl[i]        -1 -1 1 -1 3 3

nsr[i]        1 3 3 6 5 6

A  $\leftarrow$   $i - nsl[i]$   $\rightarrow$  no. of Element on left

B  $\leftarrow$   $nsr[i] - i$   $\rightarrow$  no. of Element on right

total subarray where arr[i] min = A x B

