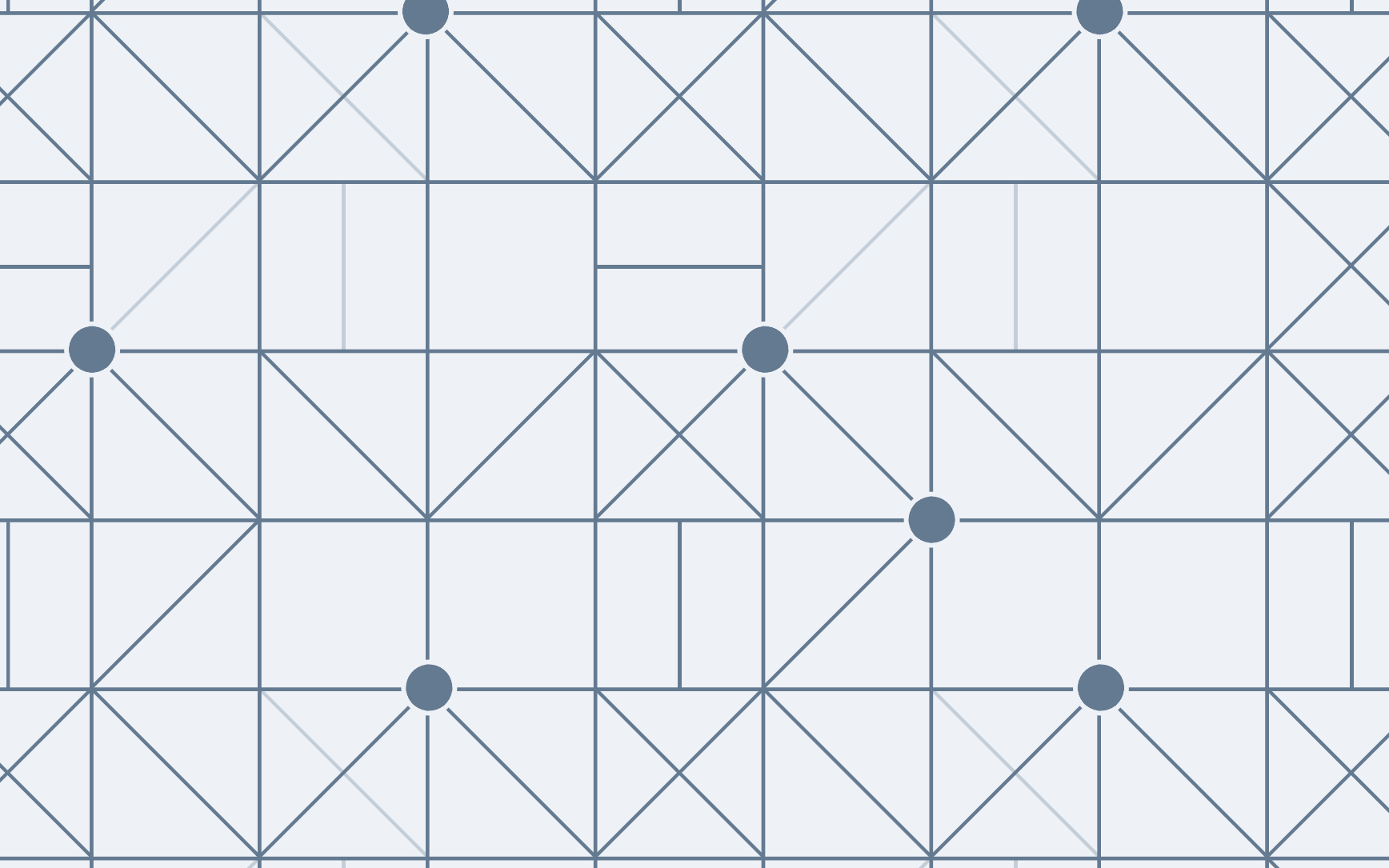
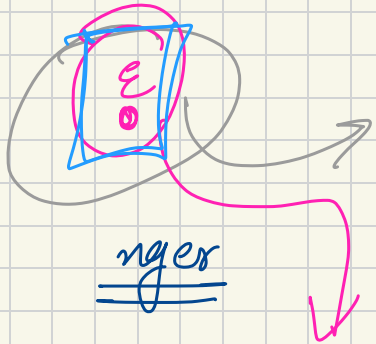


- Stacks → Basics
  → implemented.

- <u>Monotic Stack</u> → Next greater element
  → Largest Area Histogram
  → Stock Span

- infix, prefix, postfix {Conversion, Evaluation}

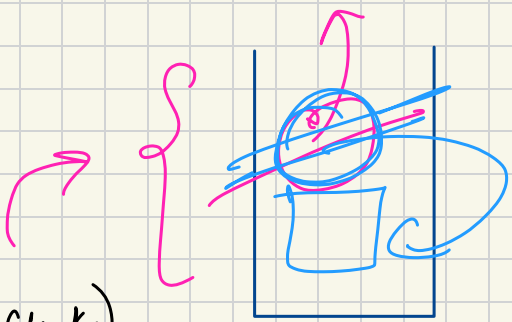
Monotonic stack

arr = {  }

nger

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

(N ele. in the stack)



Stack {people looking for nger}

move: left to right in given array.

Agenda

- Sum of Subarray minimums
- Trapping rain water
- Min. Stack

} Asked a lot in Interviews

2 lec. Queues & Stacks.

Sum of subarray Minimums.

I/P

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

Subarrays?

Subarrays:

{3} → 3	{2} → 2	{4} → 4	{1} → 1	{5} → 5	{2} → 2
{3,2} → 2	{2,4} → 2	{4,1} → 1	{1,5} → 1	{5,2} → 2	
{3,2,4} → 2	{2,4,1} → 1	{4,1,5} → 1	{1,5,2} → 1		
{3,2,4,1} → 1	{2,4,1,5} → 1	{4,1,5,2} → 1			
{3,2,4,1,5} → 1	{2,4,1,5,2} → 1				
{3,2,4,1,5,2} → 1					

O/P

Σ, subarray min ^{m} = 36

Brute force :

- Calc. all subarrays, get min value in each and get sum.

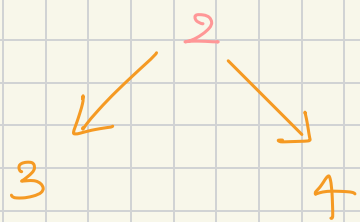
```
int sum = 0;
for (int i = 0; i < n; i++)
{
    int min = +∞;
    for (int j = i; j < n; j++)
    {
        min = Math.min(min, arr[j]);
        sum += min;
    }
}
```

TC : $O(N^2)$

SC : $O(1)$

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

Calc all the subarrays which should have this smallest element



- Principle of counting

$3 \times 2 = \boxed{6}$

arr = { 5, 3, 9, 2, 10, 4, 1 }

l r

{2}
{9, 2}
{3, 9, 2}
{5, 3, 9, 2}

$l \times 1 \times r$

2

compulsory

$\boxed{2}$

9
3, 9
5, 3, 9

l

r

10
10, 4

total Subarray

$= l \times 1 + r \times 1 + l \times 1 \times r + 1$

1

$\boxed{2}$

$l \times 1$

2

3×1

1 x

2

1×2

$\{2, 10, 8, 15, 5, 7, 3\}$

$\{5\} \longrightarrow$ Me alone.

$\{5, 7\} \longrightarrow$ just right people

$\{15, 7\}$

$\{8, 15, 7\}$

$\{10, 15, 7\}$

$\longrightarrow$ just left people.

$3 \times 1 \times 1$

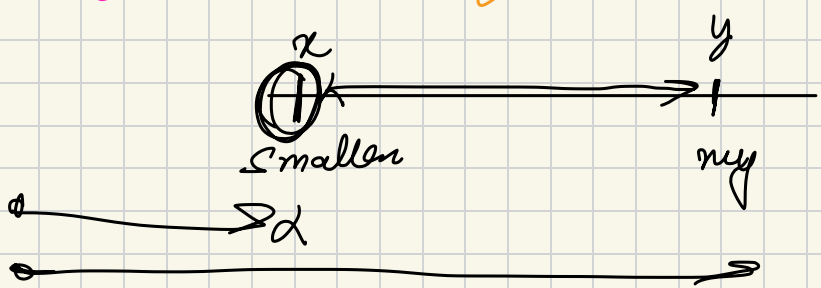
	0	1	2	3	4	5	6
arr =	5	3	9	2	10	4	1

nse li = { -1, -1, 1, $\boxed{-1}$, 3, 3, -1 }

nse ri = { 1, 3, 3, 6, 5, 6, 7 }

- $r = nseri - i - 1$ ✓

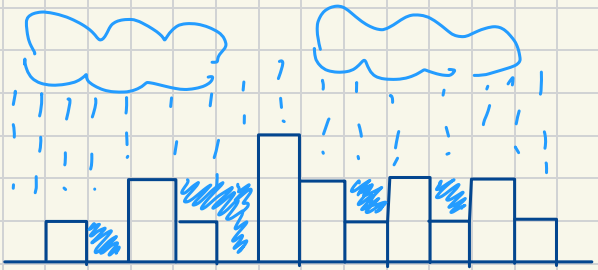
- $l \rightarrow i - nseli - 1$ ✓



$\boxed{y - x - 1}$

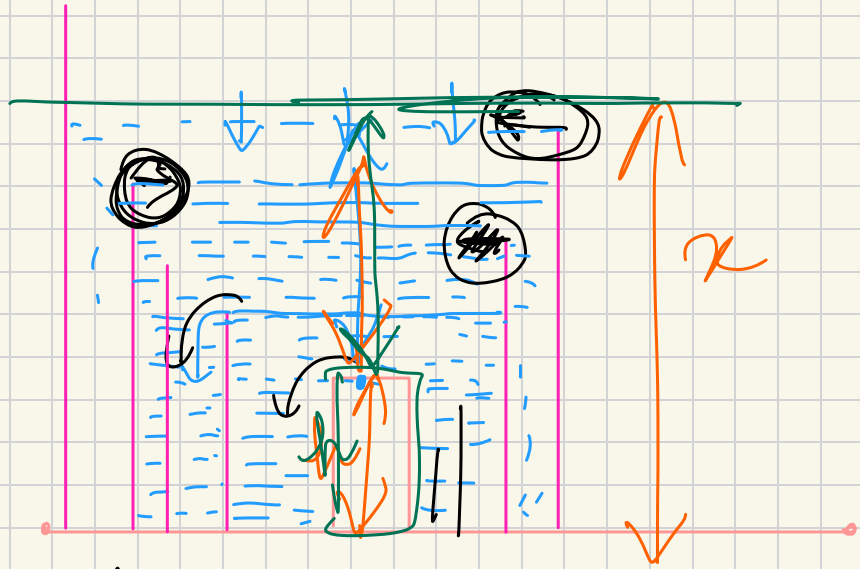
z
1

Trapping Rain Water



hist[] = { 0 1 0 2 1 0 3 2 1 2 1 2 1 0 }

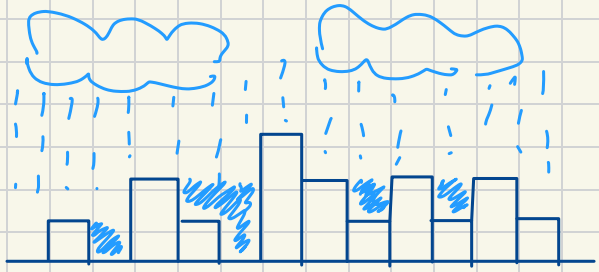
amt of water trapped = 6 sq units



min (tallest on left, tallest on right)

decides the height of water above me.

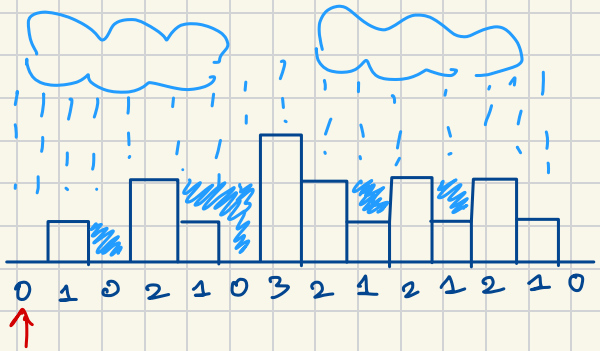
Brute force



hist[] = 0 1 0 2 1 0 3 2 1 2 1 2 1 0

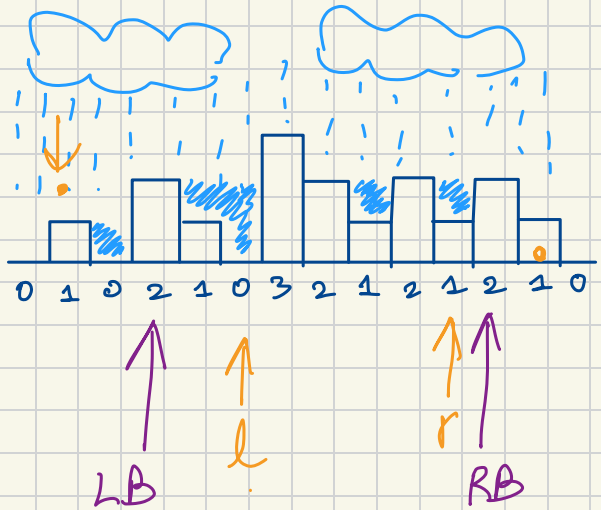
TC : $O(N^2)$
SC : $O(1)$

- go to each building
- get max height support from left and right
- Calc actual water above you { if any }



lmax = 0 0 1 1 2 2 2 3 3 3 3 3 3

rmax = 3 3 3 3 3 3 2 2 2 2 2 1 0 0



→ maxH Building of left l

LB <= RB

for l

{ support = LB;

update LB if possible

→ l++;

} else

{ support = RB;

update RB if possible

r--;

}