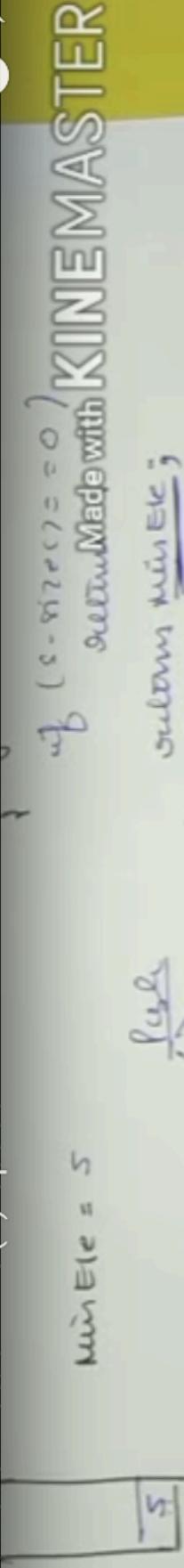


### 11 Minimum Element in Stack in O(1) Space



void push( int x )

{ if ( s.size() == 0 )

s.push(x)

minEle = x

else

? if ( x < minEle )

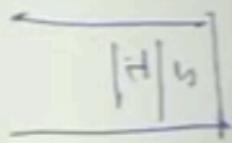
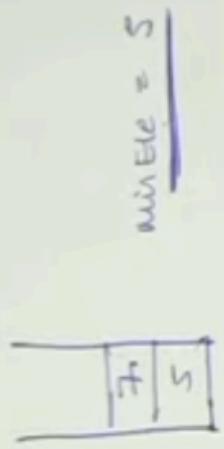
s.push(x)

(x) minEle

else

? s.push( 2\*x - minEle )

minEle = x



$$\frac{2x - \text{minEle}}{2}$$

$$2x - 5$$

$$2x - 3$$

$$2x - 1$$

### 11 Minimum Element in Stack in O(1) Space

min Ele = -1

min Ele  
Made with KINEMASTER

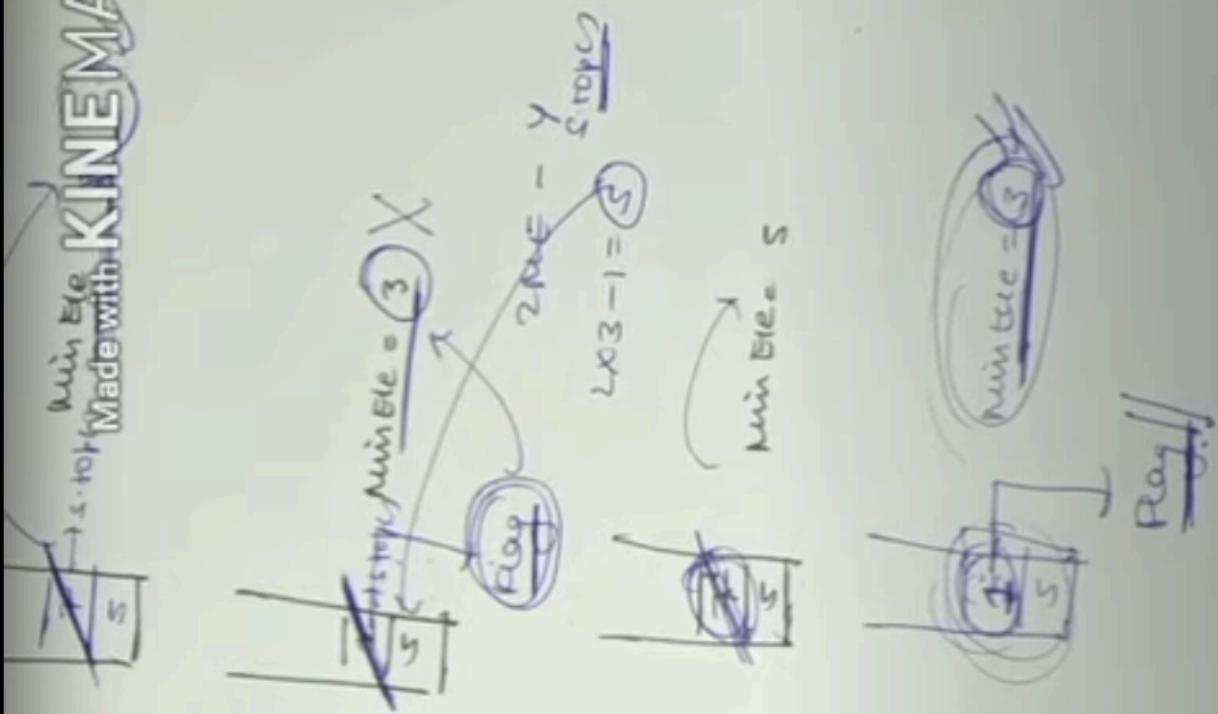
else  
{ if ( $s \cdot top() >= mg$ )  
 $s \cdot pop()$

else if ( $s \cdot top() < mg$ )  
 $mb = 2 * minEle - s \cdot top()$   
 $s \cdot pop()$

cout  $\top()$   
{ if ( $s \cdot size() == 0$ )  
 $bulletin = -1$

else  
{ if ( $s \cdot top() >= minEle$ )  
 $bulletin \cdot top() = minEle$   
else if ( $s \cdot top() < minEle$ )  
 $bulletin \cdot minEle$

{



## Stack

- 1) Nearest greater to left
- 2) Nearest greater to right →
- 3) Nearest smaller to left →
- 4) Nearest smaller to right
- 5) Stock Span Problem
- 6) Maximum Area of Histogram
- 7) Max Area of Rectangle in Binary Matrix
- 8) Rain water trapping
- 9) Implementing a min stack
- 10) Implement stack using heap
- 11) The celebrity problem
- 12) Longest valid parentheses
- 13) Iterative TOH.



1 Stack Introduction And Identification

far east  $\omega = 0$ ;  $l \in n; l++$ )

```
for( init J=0 ; J<1 ; J++ )
```

Made with KINEMASTER

$q_c$  is dependent on  $i^*$

```
for(int i = 0; i < m; i++)
```

for  $\frac{d}{dx} \tan J$

102 | a

Stack

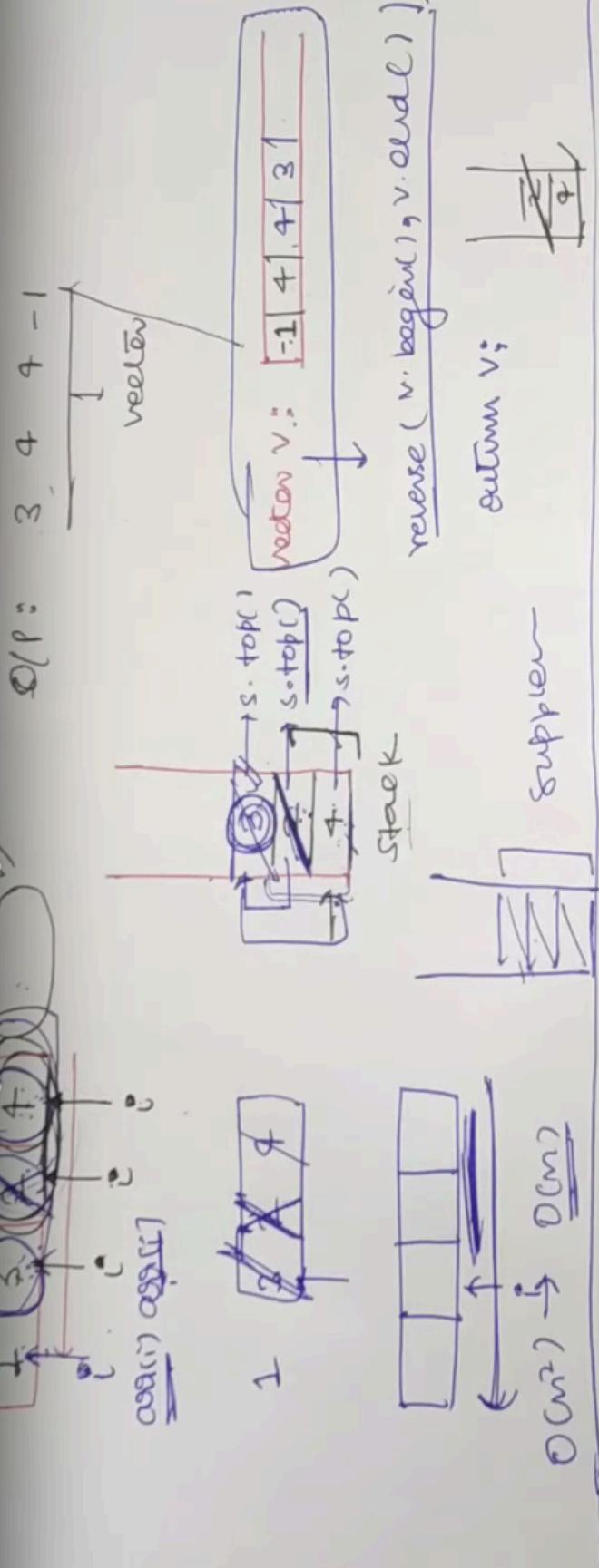
S  
↓  
S

loop

संस्कृत

14:28 / 16:08

### 2 NGR | Nearest Greater to right | Next Largest Element



- stack empty → -1
- s.top() > curr → → s.pop(); stack empty
- s.top() < curr → pop → . so top() greater than curr)

≡ 2 NGR | Nearest Greater to right | Next Largest Element

### 3 NGL | Nearest Greater to left

for (int i = size - 1; i >= 0; i--) → for (int i = 0; i < size; i++)

I

if (s.size() == 0)  
v.push\_back(-1);

else if (s.size() > 0 && s.top() > arr[i])  
v.push\_back(s.top());

else if (s.size() > 0 && s.top() <= arr[i])  
while (s.size() > 0 && s.top() <= arr[i])  
s.pop();  
if (s.size() == 0)  
v.push\_back(-1);  
else  
v.push\_back(s.top());

II

s.push(arr[i]);

### 4 NSL | NEAREST SMALLER TO LEFT

stack < dict > s;

for (int i = size - 1; i >= 0; i--) → for (int i = 0; i < size; i++)  
if (s.size() == 0)  
v.push\_back(-1);

II

Smaller

else if (s.size() > 0 && s.top() > arr[i])

v.push\_back(s.top())

II

else if (s.size() > 0 && s.top() <= arr[i])

while (s.size() > 0 && s.top() <= arr[i])  
s.pop();

→ smaller to R  
→ older to L  
→ smaller to L

if (s.size() == 0)  
v.push\_back(-1)

else  
v.push\_back(s.top());

^ (arr[i])

### ≡ 5 NSR | Nearest Smaller to Right

Stack <  $v_{left} > v_{right}$ ;

Made with KLINEMASTER

For  $l \leq i \leq r - 1$  ;  $r' = 0$  ;  $r'' = 0$  ;  $\rightarrow$  ~~for  $i = l$  to  $r - 1$~~

if  $s.size() == 0$  ;  
 $v.push\_back(-1);$

else if ( $s.size() > 0$  &  $s.top() > s[i]$ )  
 $v.push\_back(s[i]);$

else if ( $s.size() > 0$  &  $s.top() < s[i]$ )  
while ( $s.size() > 0$  &  $s.top() < s[i]$ )  
 $s.pop();$

if ( $s.size() == 0$ )  
 $v.push\_back(-1);$   
else  
 $v.push\_back(s.top());$

Smaller

Waller

Waller

Waller

Waller

Waller

### 6 Stock Span Problem

```
100 80 60 70 60 75 85
```

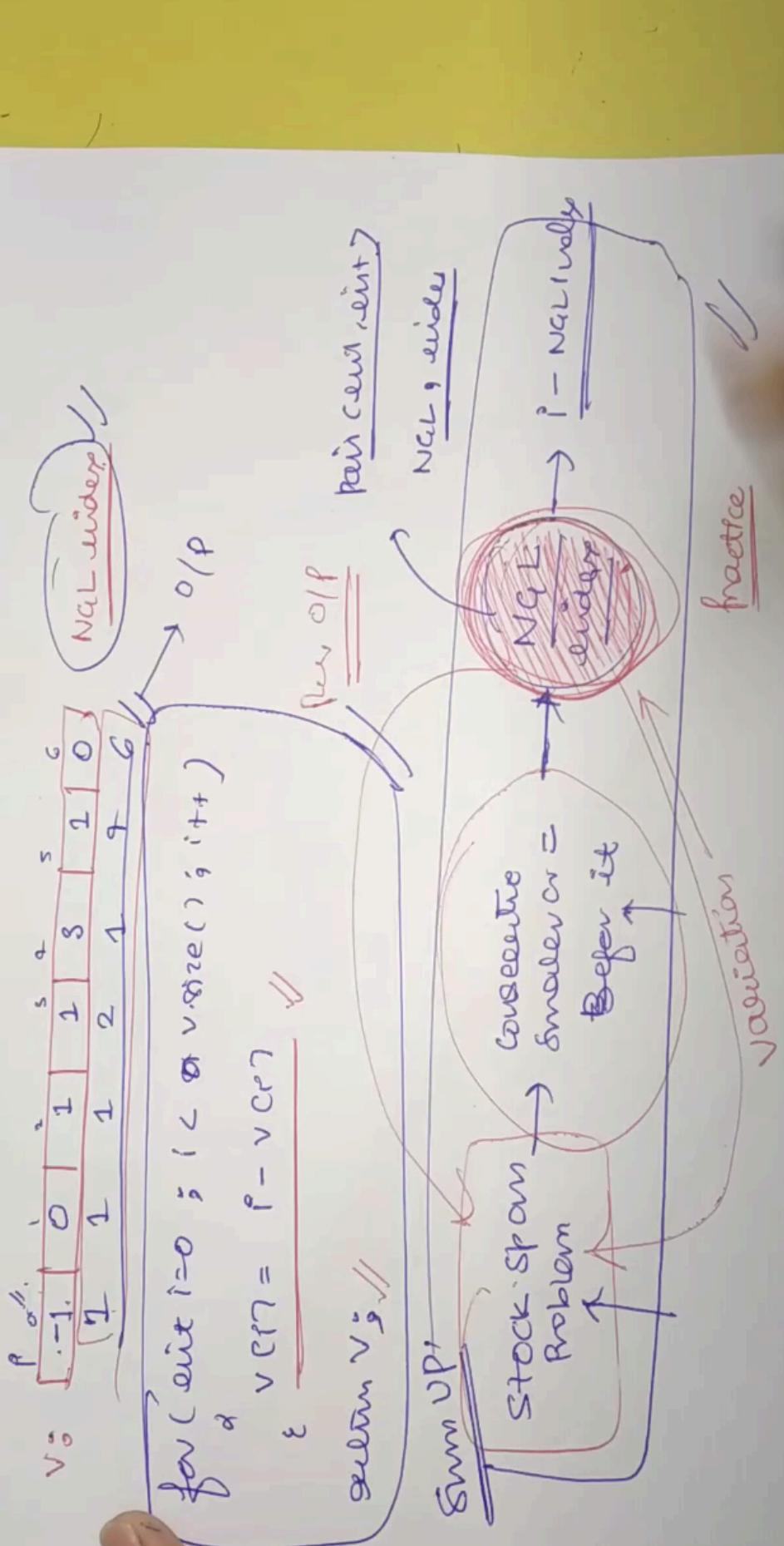
in down

NGL

$\rho_{\text{out}}$

-1	0	1	2	3	4	5	6
1	1	1	2	1	2	1	2

NGL under



7 Maximum Area Histogram | MAH

NSL Jeweler width	1	5	1	3	1	7	1
→ 6 + 2	6	2	2	5	5	7	7
→ 1 + 5	1	5	5	5	5	7	7
→ -1 + 1	-1	1	1	3	3	-1	5
→ 0 + 0	0	0	0	0	0	0	0

NCL wider

卷五

~~Cancel~~

A hand-drawn diagram of a spiral staircase. The stairs are represented by a series of concentric, roughly circular steps that wrap around a central vertical axis. The drawing is done in blue ink on white paper.

WATER = septic tank - leach field - 1

subject - left

## Babuot Gasy

length

$5 - 1 = 4 - 1 = 3$

1

6

-

10

1

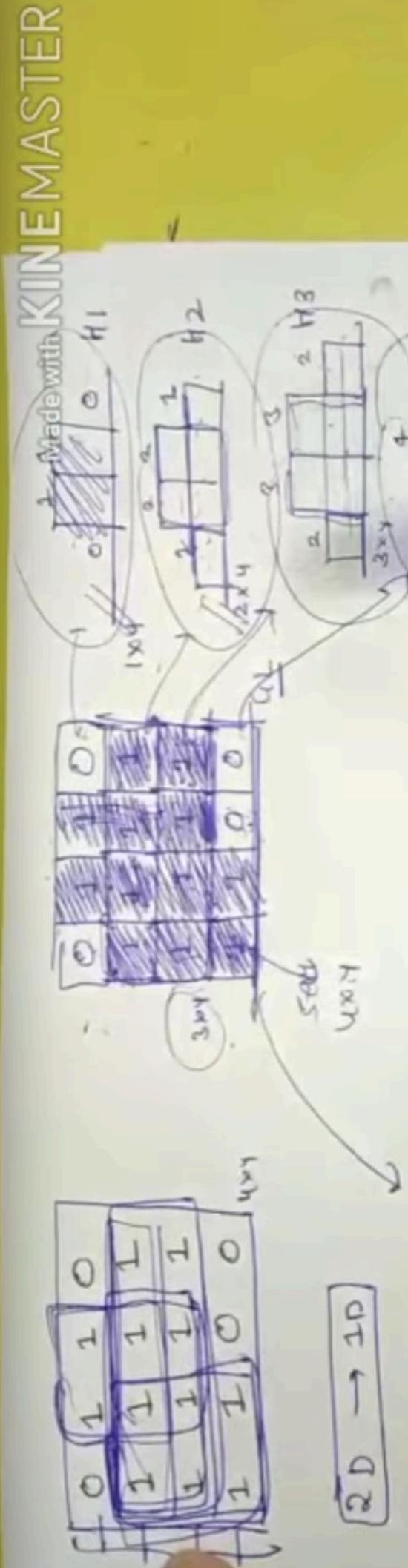
۱۰۹

9

104

◀ ▶ 🔍 20:26 / 34:59

### 8 Max Area Rectangle in binary matrix



ans = max (H1, H2, H3, H4)

max (H1) → 2  
max (H2) → 8  
max (H3) → 6  
max (H4) → 6

vector man

vector man

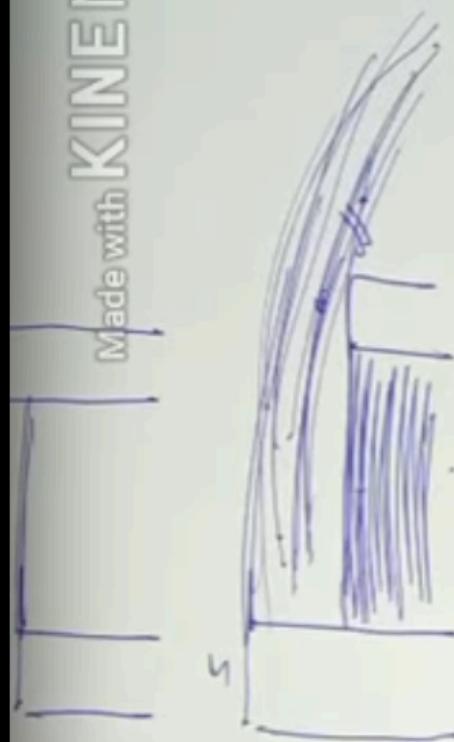
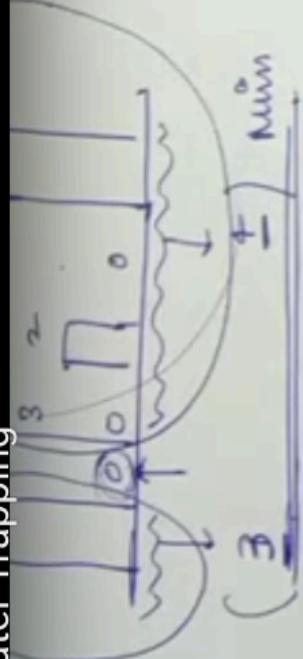
vector man

vector man

vector = @ 1 2 3 4  
vector = @ 1 2 3 4  
vector = @ 1 2 3 4  
vector = @ 1 2 3 4

0	1	1	0
1	1	1	1
1	1	1	0
1	1	0	0

### 9 Rain Water Trapping

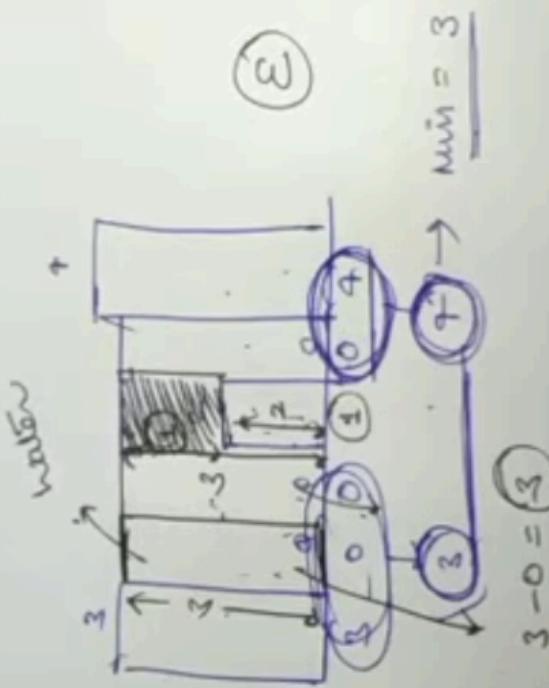


Made with KLINE MASTER

$$\text{water}[\mathcal{I}] = \min(\max[L], \max[R]) - \text{arr[I]}$$

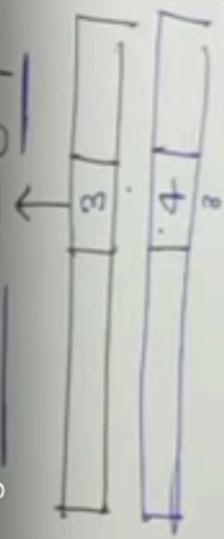
$$\frac{\text{water}[\mathcal{I}]}{\text{arr[1]=2}} = \min(3, 4) - 2 \\ = 3 - 2 = \underline{\underline{1}}$$

$$\Sigma$$

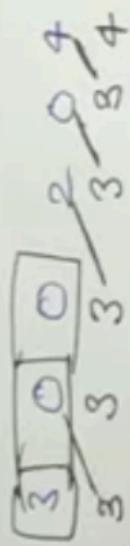


### 9 Rain Water Trapping

Made with KLINEMASTER



Water level: min [Water level, Water level] - Water level



Water level:  
Water level:

