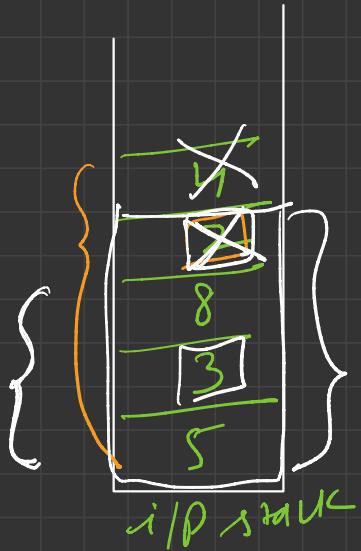



Stack

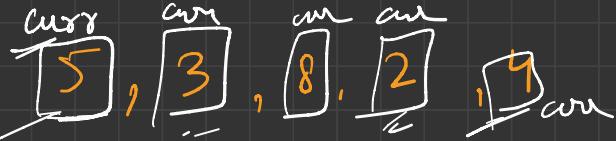
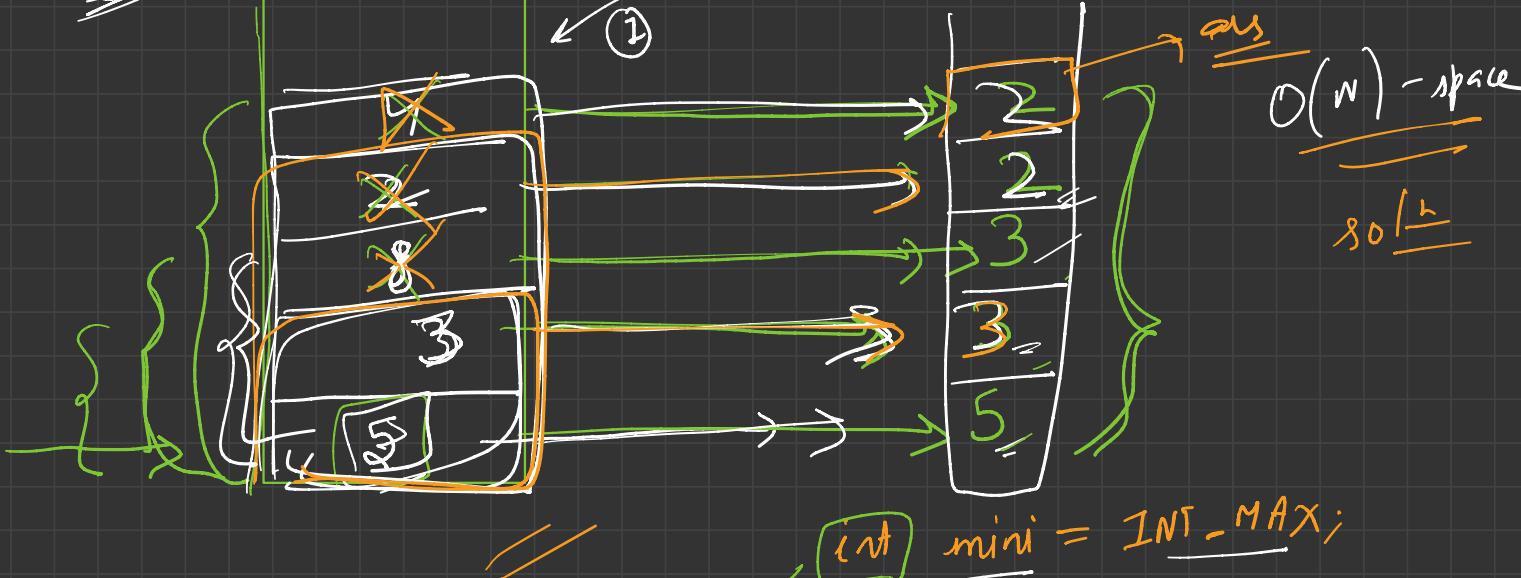
→ getMin() → $O(1)$ → T.C

$O(1)$ → S.C



getMin() → 2 → $O(1)$
pop()
getMin() → 3 →
pop()
getMin() → 3

Approach:-



①

$O(n)$ - space

so ↴

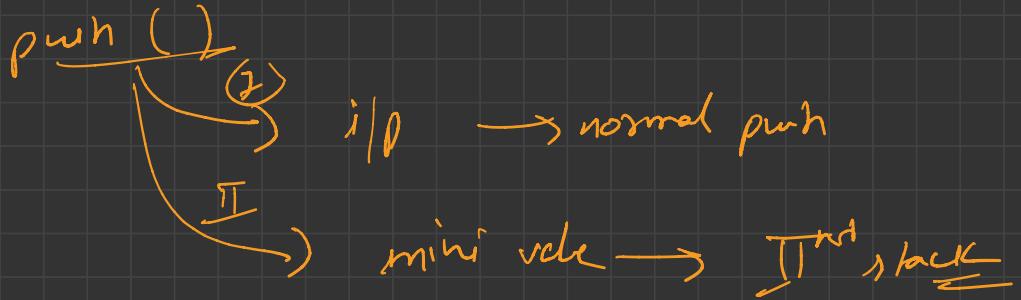
~~int~~ mini = INT-MAX;

mini = min (mini, curr)

$$= \min (5, 3) = 3$$

$$= \min (3, 8) = 3$$

$$= \min(3, 2) = 2$$
$$= \min(2, 4) = 2$$



$\text{pop}()$

I \rightarrow done stack sc top & do

$\text{getMin}()$ \rightarrow return IIth stack top

$T.C \rightarrow O(1)$

$S.C \rightarrow O(N) \rightarrow$ extra stack

→ Question

$T.C \rightarrow O(1)$

$S.C \rightarrow O(1)$

solve my

a variable

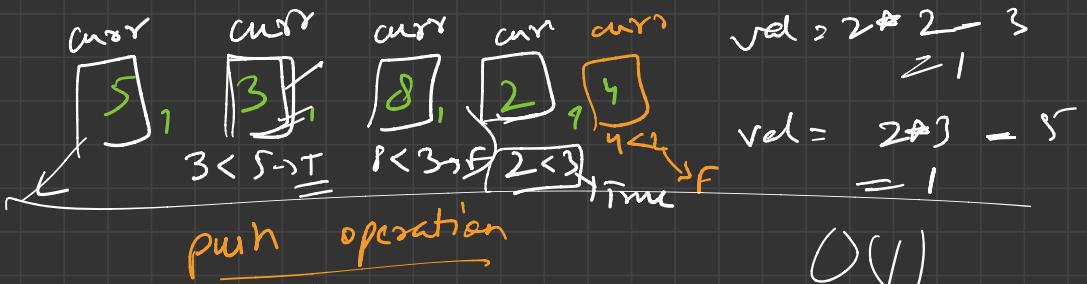
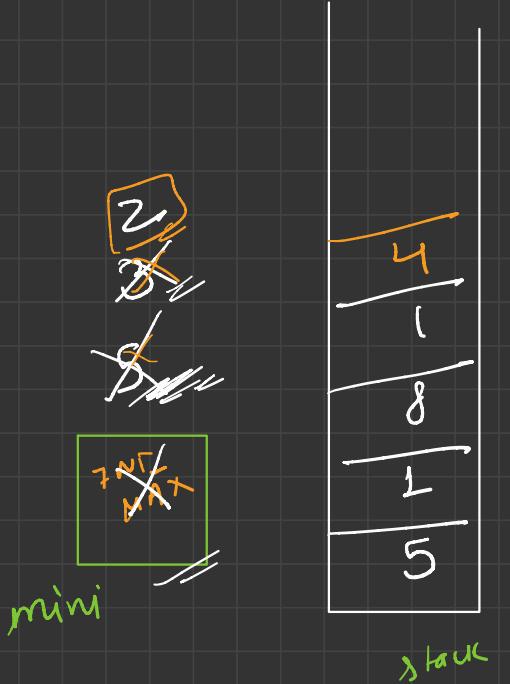
$arr[1] \rightarrow O(1)$

stack
arr
vector
set
map
D's

$\rightarrow O(N)$

int mini -

Approach #2

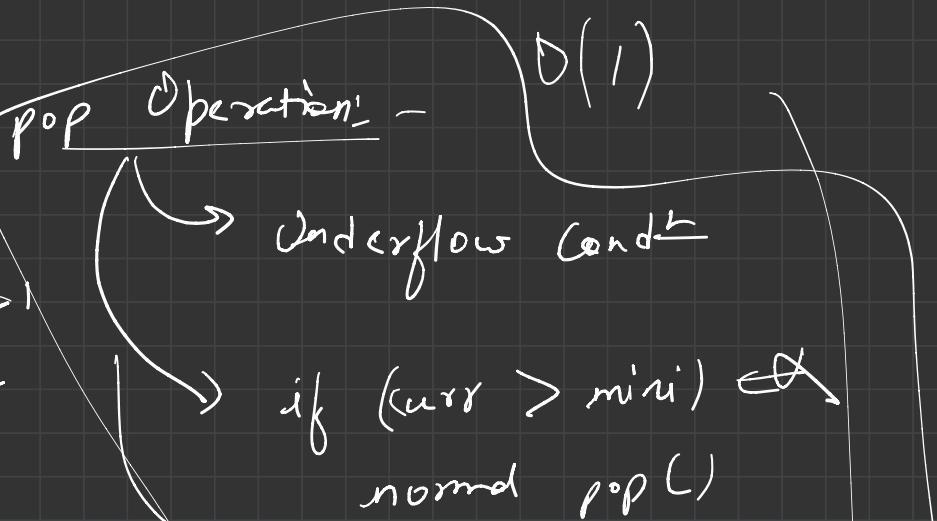
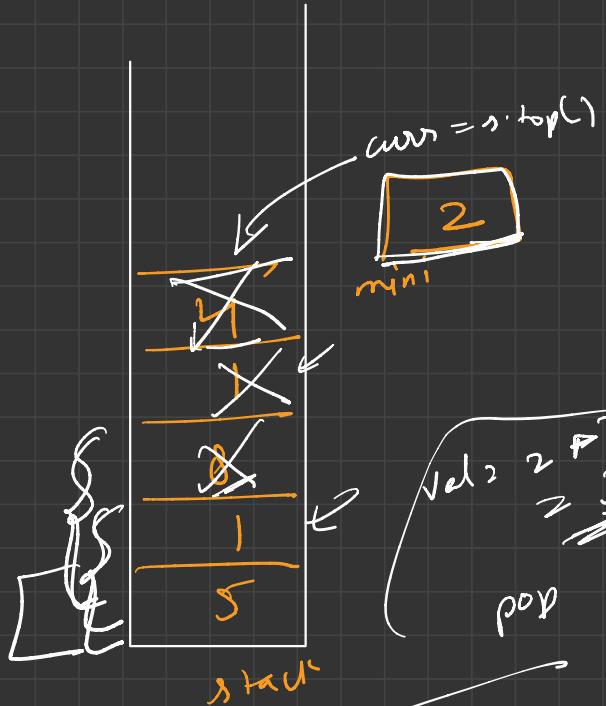


$$\begin{aligned} \text{mini} &= \overline{\min} (\text{mini}, \text{max}) \\ &= \min (\text{IN_MAX}, 5) \\ &\leq 5 \end{aligned}$$

→ back document

if (curr < mini) =
val = $2 * curr - mini$
push val
mini update

Normal push



if ($curr > mini$)
normal pop()

$$8 > 3$$

$$1 > 3 \rightarrow f$$

$$val = 2 * 3 - 1$$

$$mini = 5$$

else

$$val = 2 * mini - curr$$

$$mini = val$$

Pop()

getMin() $O(1)$

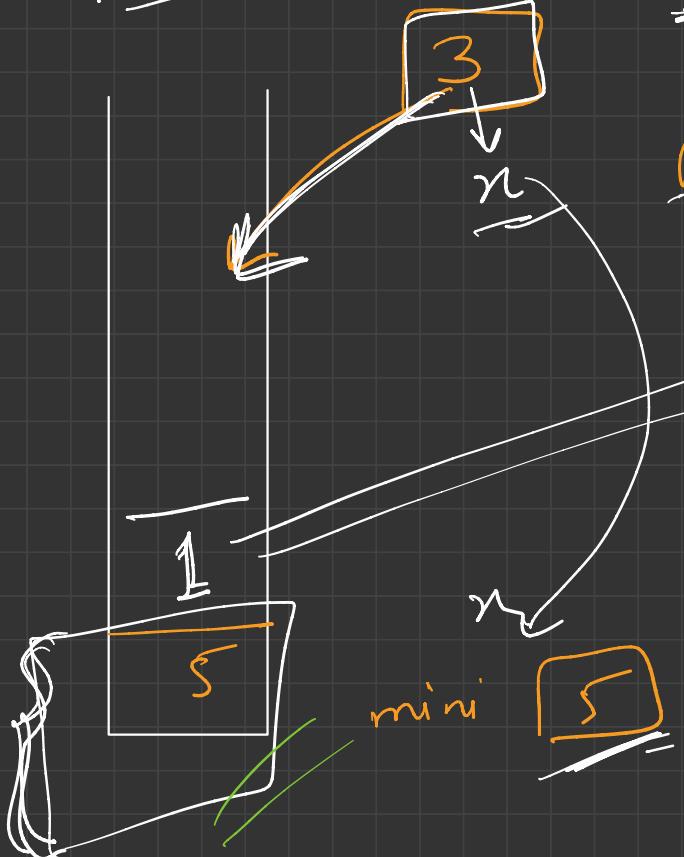
mini → stored
return

push → $2 * curr - mini$

pop() → $2 * mini - curr$

curr minimum ← previous minimum ←

Push



$$3 < 5 \rightarrow \underline{\underline{val}} = \boxed{2 * \underline{\underline{curr - min}}} \\ = 2 * 3 - 5 = 1$$

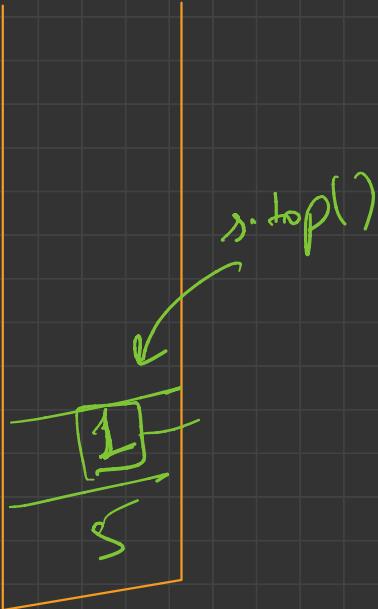
Push n

$$2 * \underline{\underline{curr - min}} \\ = 2n - \underline{\underline{min}}$$

$$\rightarrow \boxed{= 2n - \text{prevMinimum}}$$

$$\boxed{\text{min} \rightarrow n}$$

Pop()



Pop()

$$2^{\text{mini}} - \underline{\text{curS}}$$

$$2^n - \underline{\underline{\text{curS}}}$$

$$2^n - (2^n - \text{prevMin})$$

prev Minimum

25

Mom's work

